ISSN 2348-375X

Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

GENERAL FRAMEWORK TO REVERSIBLE DATA HIDING USING COEFFICIENT-BIAS ALGORITHM

Vivekanandan M[1*], Sadishkumar ST[2], Manikandaprabu N[3], Marimuthu CN[4]

[1]PG Scholar, Nandha Engineering College, TN, India
[2]Associate professor, Nandha Engineering College, TN, India
[3]PG Scholar, Nandha Engineering College, TN, India
[4]Dean & HOD, Department of ECE, Nandha Engg College, TN, India

Received: 07-12-2013; Revised: 05-01-2014; Accepted: 03-02-2014

*Corresponding Author: **M. Vivekanandan,**
PG Scholar, Nandha Engineering College Email: vivekanandan6484@gmail.com

ABSTRACT

Basically the lossless reversible data hiding schemes are able to carryover up to 256 characters only. In this case the space also considered as a character. Since low data are only able to hide in an audio signal. And there is no more proof for extraction of the data in another node after transmission through a communication channel. In a proposed system maximum of 1,000 characters are able to embed in an audio signal and a different frame work for hiding and extracting are provided at both end of the channel. A simple lossless data hiding method based on the coefficient-bias algorithm by embedding bits in both spatial domain and frequency domain is proposed. In spatial domain, each pixel in a host audio signal is first subtracted from the block-mean. Then, a stego audio signal is generated by embedding a large amount of bits (or the primary message) in the mean-removed blocks via the coefficient-bias algorithm. To provide an extra security and robustness, the stego signal is transformed to frequency domain by integer wavelet transform (IWT).The whole project is going to implement using MATLAB software.

Keywords: Reversible data hiding, Coefficient-bias algorithm, Stego audio signal, Steganography.

INTRODUCTION

Data hiding is also known as steganography (from the Greek words stegano for "covered" and graphos, "to write"). In contrast to cryptography, which focuses on rendering messages unintelligible to any unauthorized per- sons who might intercept them, the heart of steganography lies in devising astute and undetectable methods of concealing messages themselves. An obvious application is a covert communication using innocuous cover signals, like a telephone conversation or an image.

Another application, known as (digital) watermarking, refers to embedding an unobtrusive mark into an object, which can be used to identify the object. For example, a digital watermark can be inserted into a piece of music, so that radio and TV broadcasts can be monitored automatically for royalty payment purposes. Many other applications, such as piracy detection and/or prevention, proof of performance (e.g. monitoring time and duration of advertisement broad-casts), integrity verification (to detect tampering of a cover signal), traitor tracing, (e.g. to identify a source of a leak), transaction identification, auto-matic inventory, copy control, auxiliary information attachment, etc., have been reported[2-5].

The fast improvement of the Internet and the digital information revolution caused major changes in the overall culture. Flexible and simple-to-use software and decreasing prices of digital devices have made it feasible for consumers from all over the world to create, edit and exchange multimedia data. Broadband Internet connections almost an errorless transmission of data helps people to distribute large multimedia files and make identical digital copies of them. In modern communication system[6-8].

Data Hiding is most essential for Network Security issue. Sending sensitive messages and files over the Internet are transmitted in an unsecured form but everyone has got something to keep in secret. Audio data hiding method is one of the most effective ways to protect your privacy[9-12].

AIM OF THE PROJECT

We are of the belief that the easiest way to keep something from prying eyes is to place it right in front of the person looking for it and make it look as innocuous as possible.

Everyone has a taste for a certain kind of music. Hence, it is more than likely that the person will have that kind of music on the storage device of his computer. Also, it is quite common case where people share and transfer different music

files to one another. If one were able to hide the message can be. Also, transfer of this message can be done quite conveniently without raising any eyebrows[13].

Our aim is to come up with a technique of hiding the message in the audio file in such a way, that there would be no perceivable changes in the audio file after the message insertion. At the same time, if the message that is to be hidden were encrypted, the level of security would be raised to quite a satisfactory level. Now, even if the hidden message were to be discovered the person trying to get the message would only be able to lay his hands on the encrypted message with no way of being able to decrypt it[14].

STEGANOGRAPHY IN AUDIO

Data hiding in audio signals is especially challenging, because the Human Auditory System (HAS) operates over a wide dynamic range. The HAS perceives over a range of power greater than one billion to one and a range of frequencies greater than 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 which is 80dB below ambient level. However there are some 'holes' available. While it has a large dynamic range, it has a fairly small differential range. As a result, loud sounds tend to mask out the quieter sounds.

Additionally, the HAS 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.

LOW-BIT ENCODING

Low-bit encoding is the one of the simplest way to embed data into other data structures. By replacing the least significant bit of each sampling point by a coded binary string, we can encode a large amount of data in an audio signal.

Ideally, the channel capacity is 1 kb per second (kbps) per 1 kilohertz(kHz), e.g., in a noiseless channel, the bit rate will be 8 kbps in an 8 kHz sampled sequence and 44 kbps in a 44kHz sampled sequence. In return for this large channel capacity, audible noise is introduced. The impact of this noise is a direct function of the content of the host signal, e.g., crowd noise during a live sports event would mask low-bit encoding noise that would be audible in a string quartet performance figure1.1.

Adaptive data attenuation has been used to compensate this variation. The major advantage of this method is its poor immunity to manipulation. Encoded information can be destroyed by channel noise, re-sampling, etc., unless it is encoded using redundancy techniques.

In order to be robust, these techniques reduce the data rate which could result in the requirement of a host of higher magnitude, often by one to two orders of magnitude. In practice, this method is useful only in closed, digital-to-digital environments.

PROJECT DESCRIPTION

OBJECTION OF THE PROJECT

In Order to be able to define our system architecture, we must first dearly state what our objective that will derive system behavior at the same one of our objective is to create an experience, which is not only unique to the (user) client, but also makes him feel that he has loyal attachment to the system and approaches us whenever he/she needs. To achieve better results and success by implement computerized process instead of manual process.

MODULES AND THEIR DESCRIPTION

Data hiding and extracting from an audio file is done in two main modules.

- Embed module.
- Extract module.

Embed Module (To embed the text file into the audio file)

In this module, the first step is selecting an input audio file. The selection is made through opening a new dialog box and the path selected is displayed through a textbox. The second step is selecting an output audio file in which text data or a text file is embedded. The third step is choosing a text file or typing any text message for embedding. Fourth step is selecting a key file.

In the fifth step whatever the files that we have selected are viewed and verification of the path is done. In the sixth process data is embedded in to the audio file using low bit encoding technique in figure 1. After embedding the content both the audio files are played and a listener cannot find any difference between the audios.

Extract Module (To extract the text file from the audio file)

In this module, the first step is the process of selecting the encrypted audio file. This is the file that a user has to extract information from the output audio. Second process involved in selecting a new text file to display the embedded meassage. Symmetric encryption method is used here, so the key selected during the embedding process is used in decrypting the message. All the process done till now are displayed using a list box and finally the embedded message can be viewed with the help of a file or in a textbox.

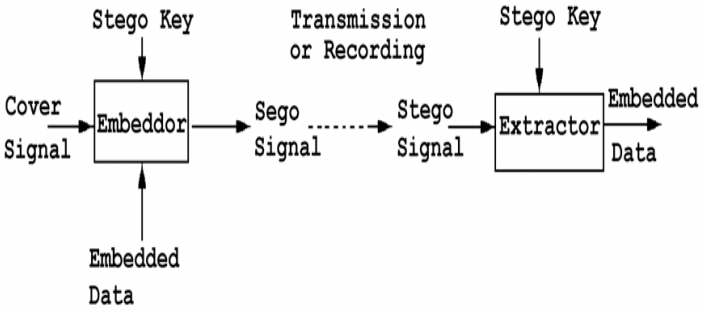

Figure 1: Block diagram of encoding and decoding

COEFFICIENT-BIAS ALGORITHM

The idea of the coefficient-bias algorithm is to embed data bits in both spatial domain and frequency domain. That is, a stego signal is first generated by embedding the primary message in the spatial domain. Then, the stego-signal is decomposed to IWT domain for hiding the secondary Stego key. The schematic view of the proposed method is depicted in Fig.2.2. Some notations shown in the figure are defined in the following:

- A audio signal
- contains a secret message
- Contains stego key.

- The IWT domain obtained from encrypted signal.
- A mixed signal contains a secret message and a stego key.

Consequently, both the stego key and the secret message would be lossless extracted and the host signal are perfectly restored at receiver site. The details of the coefficient-bias algorithm are specified in the following sections.

DATA HIDING

General principles of data hiding technology, as well as terminology adopted at the First International Workshop on Information Hiding, Cambridge, U.K. A data message is hidden within a cover signal (object) in the block called embeddor using a stego key, which is a secret set of parameters of a known hiding algorithm. The output of the embeddor is called stego signal (object). After transmission, recording, and other signal processing which may contaminate and bend the stego signal, the embedded message is retrieved using the appropriate stego key in the block called extractor.

A number of different cover objects (signals) can be used to carry hidden messages in Figure 2.2. 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. 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 “holes” 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. Now we will discuss many of these methods of audio data hiding technology.

PREVIOUS WORKS

This section presents some common methods used for hiding secret information in audio.

Many software implementations of these methods are available on the Web and are listed in the relatives section. Some of the latter methods require previous knowledge of signal processing techniques, Fourier analysis, and other areas of high level mathematics. When developing a data-hiding method for audio, one of the first considerations is the likely environments the sound signal will travel between encoding and decoding. There are two main areas of modification which we will consider. First, the storage environment, or digital representation of the signal that will be used, and second the transmission pathway the signal might travel[4].

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[5]. Figure 2, shows the parity coding procedure.

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[4].

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[5].

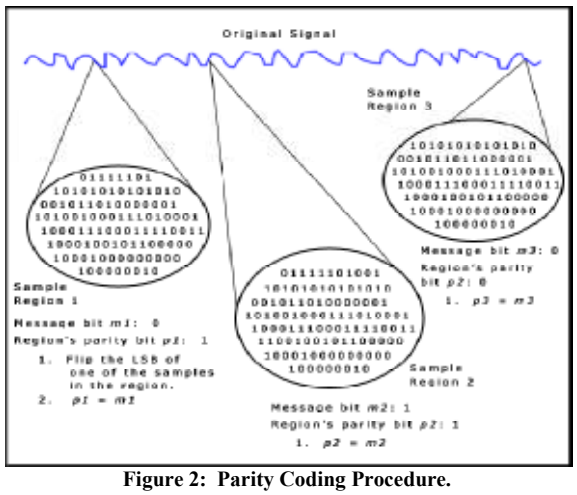

Figure 2: Parity Coding Procedure.

Phase coding is explained in the following procedure:

a. The original sound signal is broken up into smaller segments whose lengths equal the size of the message to be encoded.
b. A Discrete Fourier Transform (DFT) is applied to each segment to create a matrix of the phases and Fourier transform magnitudes.
c. Phase differences between adjacent segments are calculated.

d. 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:
e. A new phase matrix is created using the new phase of the first segment and the original phase differences.
f. 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 Figure 3 for phase cosing procedure).

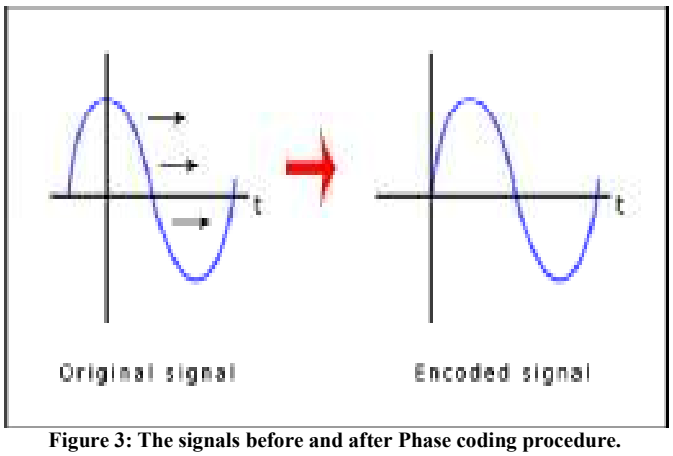

Figure 3: The signals before and after Phase coding procedure.

Spread Spectrum

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[15].

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 5).

For decoding, phase values f0 and f0 + p are interpreted as a "0" or a "1," which is a coded binary string [4].

In the decoding stage, the following is assumed:

a. The pseudorandom key is maximal (it has as many combinations aspossible and does not repeat for as long as possible). Consequently it has a relatively flat frequency spectrum.

b. 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.

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

Echo Hiding

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like the spread spectrum method, it too provides advantages in that it allows for a high data transmission rate and provides superior robustness when compared to the noise inducing methods. If only one echo was produced from the original signal, only one bit of information could be encoded. Therefore, the original signal is broken down into blocks before the encoding process begins. Once the encoding process is completed, the blocks are concatenated back together to create the final signal[16]. Echo Hiding is shown in Figure 6. Also, a message can be encoded using musical tones with a substitution scheme.

CONCLUSION

In this paper I have introduced a self bias 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. 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.

REFERENCES

1. Alattar.M, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process., 2004; 13 (8): 1147–1156.

2. Caldelli R, Filippini F and Becarelli R, Reversible watermarking techniques: An overview and a classification, Eur. Assoc. Signal Process. J. Inf. Security, 2010; 2: 1–19.
3. Celik MU, Sharma G, Tekalp AM and Saber E, Lossless generalized-LSB data embedding, IEEE Trans. Image Process., 2005; 14 (2): 253–266.
4. Coltuc D and Chassery JM, Very fast watermarking by reversible contrast mapping," IEEE Signal Process. Lett., 2007; 14 (4) 255–258.
5. Coltuc D, Low distortion transform for reversible watermarking, IEEE Trans. Image Process., 2012; 21 (1): 412–417.
6. Fridrich J, Goljan M and Du R, Lossless data embedding-new paradigm in digital watermarking," Eur. Assoc. Signal Process. J. Appl. Signal Process., 2002; 2:185–196.
7. Gao XL. An Yuan Y, Tao D and Li X, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol., 2011; 21 (8): 1061–1070.
8. Hu Y, Lee HK and Ji L, DE-based reversible data hiding with improved overflow location map," IEEE Trans. Circuits Syst. Video Technol., 2009; 19 (2): 250–260.
9. Kamstra L and Heijmans.HJAM, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process., 2005; 14 (12): 2082–2090.
10. Luo L, Chen Z, Chen M, Zeng X, and Xiong Z, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forens. Security, 2010; 5 (1): 187–193.
11. Li X, Yang B, and Zeng T, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process., 2011; 20 (12): 3524–3533.
12. Peng F, Li X, and Yang B, Adaptive reversible data hiding scheme based on integer transform," Signal Process., 2012; 92 (1): 54–62.
13. Thodi DM and Rodriguez JJ, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process., 2007; 16 (3): 721–730.
14. Tian J, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol., 2003; 13 (8): 890–896.
15. Tai WL, Yeh CM and Chang CC, Reversible data hiding based on histogram modification of pixel differences," IEEE Trans. Circuits Syst. Video Technol., 2009; 19 (6): 906–910.
16. Weng S, Zhao Y, Pan JS and Ni R, Reversible watermarking based on invariability and adjustment on pixel pairs, IEEE Signal Process. Lett., 2008; 15 (11): 721–724.

Source of support: Nil, Conflict of interest: None Declared