International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011
Proceedings published by International Journal of Computer Applications® (IJCA)

Data Hiding in Multimedia Audio using VLSI technology

Nithya A
PG student (VLSI Design),
Adhiparasakthi Engineering College,
(Affiliated to Anna University, Chennai),
Melmaruvathur,India.

Gnanasekar A K
Assistant professor,
Adhiparasakthi Engineering College,
(Affiliated to Anna University, Chennai),
Melmaruvathur,India.

ABSTRACT

This paper is devoted to VLSI technology for high security purpose. Textual information is embedded in to an audio using LSB coding and it is encrypted and decrypted by Steganography algorithm. Low bit encoding or Least Significant Bit (LSB) encoding technique is proposed to encode the least significant bit of a host file with a bit of the Steganography data. This method assumes that this alteration introduces only a minutiae difference into the host file which would be hard to detect. Low-bit encoding replaces the least significant bit of information in each sampling point with a coded binary string. This me t h o d can be efficiently employed to encode large amount of hidden data in a given audio signal. Low-bit encoding embeds secret data into the least significant bit (LSB) of the audio file. The channel capacity is 1KB per second per kilohertz (44 kbps for a 44 KHz sampled sequence). This method is easy to incorporate in to audio file by Steganography. Here all types of audio files such as uncompressed, lossless compressed, loss compressed audio can be used. This method has very high security with low expensive. A Verilog coding is used in VLSI technology.

Keywords: **Data hiding, LSB coding, Steganography, Audio file, Text data.**

1. OVERVIEW

General Principles of data hiding technology, as well as terminologyvadopted at the First international Workshop on Information Hiding,vCambridge, U.K. [1] are illustrated in Figure 1. A data message isvhidden within a cover signal (object) in the block called embeddorvusing a stego key, which is a secret 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 [2].

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 the presence of a loud signal (masker), another weaker signal may be inaudible, depending on spectral and temporal characteristics of both masked signal and masker [3]. Masking models are extensively studied for perceptual compression of audio signals [2]. 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 (80dB 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 [4].

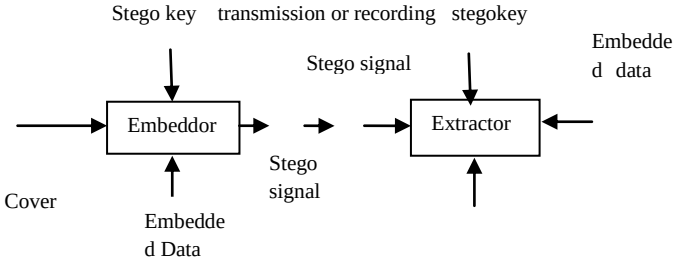

Figure 1.Diagram of data hiding and Retrival

2. RELATED WORKS

This section presents some common methods used for hiding secret information in audio. Many software implementations of these methods are available on web and are listed in the relative section. 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. Storage environment and transmission pathway are the two main area of modification which will consider [4]. *Wet paper codes* One of the prior works in audio data hiding technique is Wet paper codes. This type of code is used in MPEG audio. Its information-theoretical model is similar to the communication channel: writing in memory with defective cells. In wet paper codes, sender and receiver share only two items: a public function and a secret stego key. But this method is not suitable for all types of audio files [6].*Phase coding* Phase coding method works by substituting the phase of an initial audio

segment with reference phase that represents the data. When the phase relation between each frequency component is dramatically changed, noticeable phase dispersion will occur. *Parity coding* Instead of breaking signal down into separate regions of samples and encodes each bit from the secret message in a sample region's parity bit [5].*Echo hiding* In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. If only one echo was produced from original signal, only one bit of information could be encoded. Therefore, the original signal is broken down into blocks before the encoding process begins. A normal musical piece can be selected together with an encoding scheme that will represent a message [7, 8].

3. PROPOSED WORK

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 bit stream with a binary message, LSB coding allows for a large amount of data to be encoded. The LSB encoding involves the following steps: Receives the audio file in the form of bytes and convert it into bit pattern. Each character in the message is converted into bit pattern. Replaces the LSB bit from audio with LSB bit from character in the message.

This system provides a good, efficient method for hiding the data from hackers and sent to the destination in a safe manner. LSB encoding will not change the size of the file even after encoding and also suitable for any type of audio file format. Encryption and Decryption techniques have been used to make the security system robust. Low-bit encoding embeds secret data into the least significant bit (LSB) of the audio file. The channel capacity is 1KB per second per kilohertz (44 kbps for a 44 KHz sampled sequence). This method is easy to incorporate.

4. BLOCK DIAGRAM AND ITS DESCRIPTION

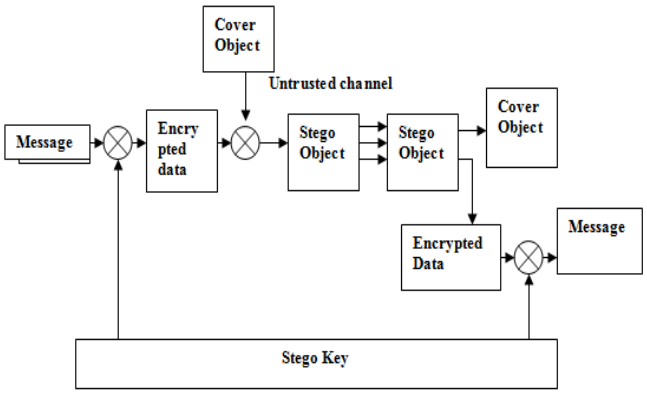

Figure.2 Block Diagram

The message that is, original data that has to be hidden is embedded into a harmless object like audio, video etc., which is defined as the cover-object. In this project we use audio as cover-object as shown inFig.1.The message is then embedded into an audio file, generally with use of a stego-key. This stego- key is generated using Linear Feedback shift register. Linear Feedback shift register is one of a pseudorandom generator. The LFSR is a shift register that has some of its outputs together in exclusive-OR configurations to form a feedback path. LFSRs are frequently used as pseudorandom pattern generators to generate a random number of 1s and 0s. The resulting encrypted message is then embedded into the cover-object, which results in stego-object. Ideally the stego- object is indistinguishable from the original message, appearing as if no other information has been encoded. The cover object is only used for the stego-object generation and is then discarded. The hope of the system is that the stego-object will be close enough in appearance and statistics to the original such that the presence of information will go undetected.

5. DATA HIDING

5.1 Audio File Format

An audio file format is a file format for storing audio data on a computer system. It can be a raw bit stream, but it is usually a container format or an audio data format with defined storage layer. The general approach towards storing digital audio is to sample the audio voltage which, on playback, would correspond to a certain level of signal in an individual channel with a certain resolution the number of bits per sample in regular intervals (forming the sample rate). This data can then be stored uncompressed, or compressed to reduce the file size. There are three major groups of audio file formats such as Uncompressed audio formats, such as WAV, AIFF, AU or raw header-less PCM, formats with lossless compression, such as FLAC, Monkey's Audio (filename extension APE), Shorten, TTA, ATRAC Advanced Lossless, Apple Lossless, MPEG-4 SLS, MPEG-4 ALS, MPEG-4 DST, Lossless, formats with lossy compression, such as MP3, Vorbis, Musepack, AAC, ATRAC and lossy Windows Media Audio (WMA).All these type of audio files are used.

5.2 Wave File Format

The WAVE file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. A RIFF file starts out with a file header followed by a sequence of data chunks. A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks -- an "fmt" chunk specifying the data format and a "data" chunk containing the actual sample data. Call this form as "canonical form".

5.3 Manipulation of Text Data

5.3.1 Pseudorandom Number Generator

A pseudorandom number generator (PRNG) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state. Although sequences that are closer to truly random can be generated using hardware random number generators, pseudorandom numbers are important in practice for simulations (e.g., of physical systems with the Monte Carlo method), and are central in the practice of cryptography and procedural generation. Common classes of these algorithms are linear congruential generators, Lagged Fibonacci generators, linear feedback shift registers, feedback with carry shift registers, and generalized feedback shift registers.

5.3.2Periodicity

A PRNG can be started from an arbitrary starting state using a seed state. It will always produce the same sequence thereafter when initialized with that state. The maximum length of the sequence before it begins to repeat is determined by the size of the state, measured in bits. However, since the length of the maximum period potentially doubles with each bit of 'state' added, it is easy to build PRNGs with periods long enough for many practical applications. If a PRNG's internal state contains n bits, its period can be no longer than 2n results. For some PRNGs the period length can be calculated without walking through the whole period. Linear Feedback Shift Registers (LFSRs) are usually chosen to have periods of exactly 2n−1.

5.3.3 Linear Feedback Shift register

Linear feedback shift registers make extremely good pseudorandom pattern generators. When the outputs of the flip- flops are loaded with a seed value (anything except all 0s, which would cause the LFSR to produce all 0 patterns) and when the LFSR is clocked, it will generate a pseudorandom pattern of 1s and 0s. Note that the only signal necessary to generate the test patterns is the clock. LFSRs have long been used as pseudo- random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very uniformly distributed output streams. However, an LFSR is a linear system, leading to fairly easy cryptanalysis. For example, given a stretch of known plaintext and corresponding cipher text, an attacker can intercept and recover a stretch of LFSR output stream used in the system described, and from that stretch of the output stream can construct an LFSR of minimal size that simulates the intended receiver by using the Berlekamp-Massey algorithm. This LFSR can then be fed the intercepted stretch of output stream to recover the remaining plaintext. LFSR is shown in Fig.3

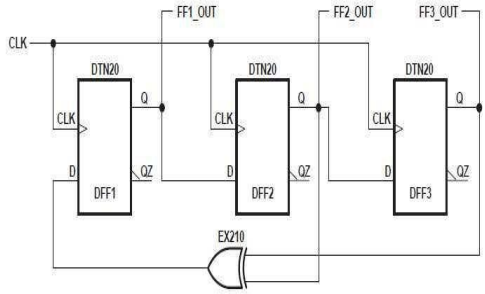

Figure3. Linear Feedback Shift Register

5.4 Encryption

Encryption is the conversion of data (plain text) into a form called a cipher text that cannot be easily understood by unauthorized people. In this project the original data that is to be hidden in a cover object is encrypted. For the purpose of encryption a key is generated using Pseudo Random key stream generator (LFSR) and this process is shown in Fig.4.

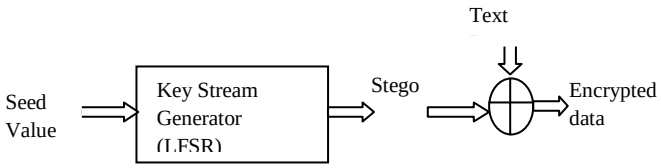

Figure 4. Encryption Block Diagram

5.5 LSB Substitution

The least significant bit of a sample is replaced with a message bit. 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. The LSB substitution is shown below with an example.

Sampled audiostream(16bit)	HEY	message encoded
1001101011011110	0	1001101011011110
1111101011000100	1	1111101011000101
0010101100101101	0	0010101100101100
0110001010100100	0	0110001010100100
1010011010010111	1	1010011010010111
0101001001010101	0	0101001001010100
0011010010100000	0	0011010010100000
0101010011010011	0	0101010011010010

6 DATA EXTRACTION

6.1 LSB Extraction

In the LSB extraction process the least significant bit of each audio stream is extracted to get the encrypted form of original information. 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.

6.2 Decryption of Extracted Data

The LSB of audio was extracted at the receiver end to get the information to be decrypted. Decryption is the process of getting the encrypted information back to its original form which is shown in Figure 5.

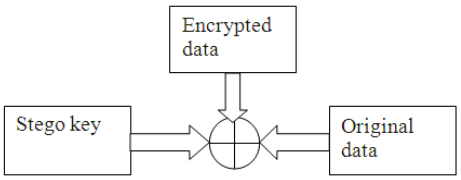

Figure 5.Decryption of Extracted Data

7.RESULTS

Here Hex Editor Neo software was used to convert audio file into a binary. Model Sim6.4 is used to obtain the simulation result and the coding's for encryption and decryption are written in Verilog. The coding's implemented in an Altera D2E kit.Audio files are converted into binary as well as the hexadecimal format by using Hex editor neo software. Here Fig .6 shows an audio file converted into a hexadecimal format. Here the input audio file is int32WE-AF which is a sample music.

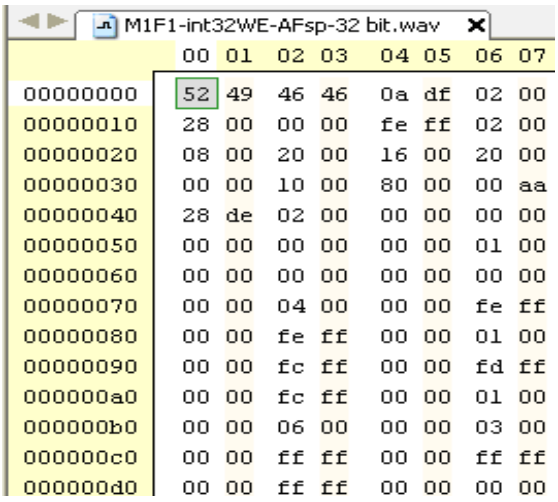

Figure.6 Audio to hexadecimal converted

After converting the audio to hexadecimal format, the next step is to hide the text data into the audio. The text data which is converted into a hexadecimal is hided in to the audio. It is represented by the gray shaded region is shown in Fig.6.

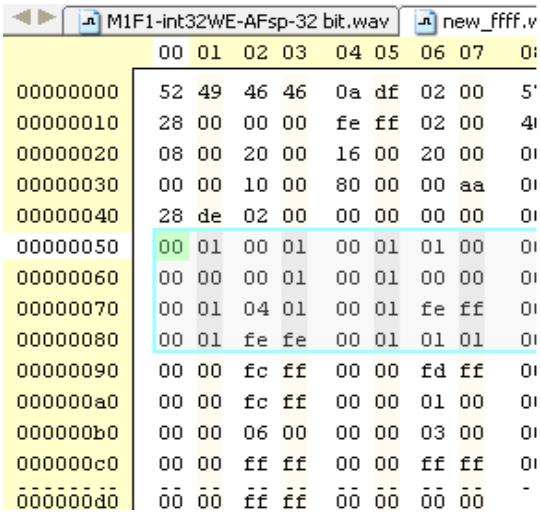

Figure.7 Audio with hidden data

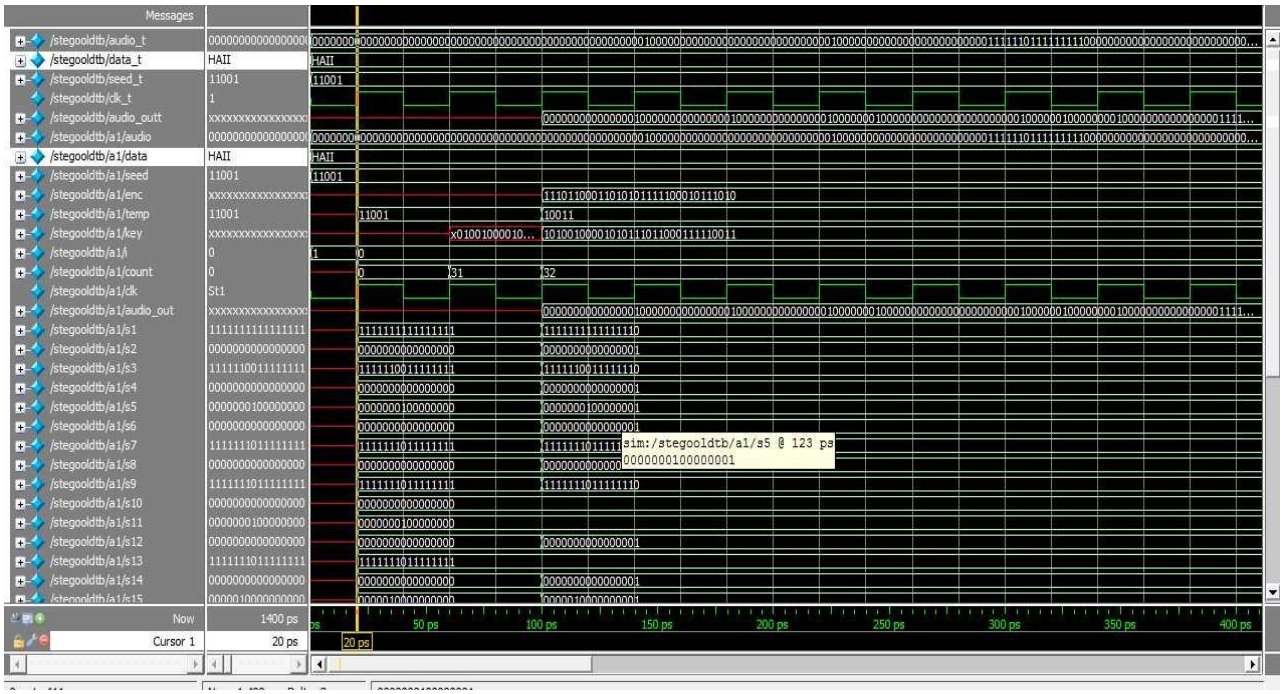

Figure 8.SimulationOutput

In this the input data taken is "HAII" which is converted into binary format, and then the audio signal is converted in to a binary which is taken as 16-bit samples. Then by using LSB substitution the data is encrypted. Again the data is decrypted using the stego key. This is shown in Fig.8.

9. CONCLUSION

Thus 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 this 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. Audio file is manipulated in a way that may be detected by the receiver with a proper key by VLSI technology.

10. REFERENCES

[1] B.Pfitzmann " Information Hiding Terminology", First International Workshop on Information Hiding, May 30-June 1,1996, Cambridge, UK, pp.347-350.

[2] Rade Petrovi, Kanaan Jemili, Joseph M. Winogard, Illija Stojanovi, Eric Metosis, " DATA HIDING AUDIO SIGNALS", June 15, 1999, MIT Media Lab, Series: Electronics and Energetic vol. 12, No.2, pp 103-122.

[3] J.Johnston and k.Brandenburg, "Wideband Coding Percaptual Consideration for speech and Music". Advances in Speech Signal Processing, S. Furoi and M.Sondhi, Eds. New York: Marcel Dekker, 1992.

[4] W.Bender, W. Butera, D.Gruhl,r.Hwang,F.J. Paiz, S. Pogreb, " Techniques for Data hiding", IBM systems Journal, Volume 39, Issue 3-4, July 2000,pp-547 – 568.

[5] Samir Kumar Bandyopadhyay, Debnath Bhattacharyya, Poulami Das, Debashis Ganguly and Swarenedu Mukherjee," A tutorial review on Steganography", " International Conference On contemporary Computing (IC3-2008), Noida, India, August 7-9, 2008,pp105 - 114.

[6] Xiaomei Quan , Hongbin Zhang , " Data Hiding in MPEG Compressed Audio Using Wet Paper Codes" 2006 IEEE, International Conference on Pattern Recognition.

[7] Robert Krenn, " Steganography and Steganalysis", An Article, January 2004.http://www.krenn.nl/univ/cry/steg/article.pdf

[8]Francesco Quirolo, " Steganography in Images", Final CommunicationReport.
http://eric.purpletree.org/file/Steganograohy%20In%20 Images.pdf.

[9] Poulami Dutta, DebnathBattacharya, "Data Hiding in Audio Signal: A Review" 2009 IEEE, International Journal of Database Theory and ApplicationVol.2.