Anusandhanvallari
Vol 2023, No.1
June 2023
ISSN 2229-3388

Strategic Management of Audio Steganography: LSB Modification Techniques for Secure Communication analysis

[1] Prof. Vaishali Nikam, [2]Prof.Mahesh Mahankal, [3]Dr. Sachin Misal, [4]Dr. Bade Ashwini Vivekanand*

[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor
[1]Affiliation Address: SSMS's Institute of Management and Research, Pune, India
[2]Affiliation Address: International Institute of Management Science, Chinchwad, Pune, India
[3]Affiliation Address: International Institute of Management Science, Chinchwad, Pune, India
[4]Affiliation Address: Siddhant College of Engineering, Pune, India

Abstract

This paper outlines a strategic approach to managing secure data transmission through audio steganography, leveraging the Least Significant Bit (LSB) modification technique to embed text-based information within host audio files. Effective management of audio steganography enables enhanced security and capacity for data hiding, surpassing text and image steganography methods. The proposed methodology involves sampling the audio file, identifying optimal embedding points, and modifying specific bits to conceal textual data without compromising audio quality. A robust management framework is presented for embedding text within audio files, accommodating diverse music genres. The resulting steganographed audio file exhibits minimal distortion, ensuring secure communication through email transmission. A MATLAB-based application is developed to facilitate encoding and decoding of secret messages, providing a reliable management tool for authorized personnel to exchange confidential information. The primary objective of this research is to establish a secure communication management system, safeguarding data exchange between sender and receiver.

Keywords- Audio steganography, secure communication, LSB modification.

I. Introduction

In today's competitive landscape, safeguarding sensitive information from unauthorized access is crucial. To address this need, a novel approach to audio steganography is proposed, enabling the concealment of data within host audio files without introducing noticeable alterations. Steganography, derived from the Greek term "covered writing," specializes in hiding data to prevent detection. By employing LSB modification, text is embedded into the audio file, creating an imperceptible change. The technique involves replacing the LSB of binary equivalents of selected audio samples with the binary representation of the secret message. A custom program reads the audio file bit by bit, storing the modified data in a separate output file. For instance, embedding the word "Secret" requires inserting its binary values into the audio file. An algorithm is developed to modify multiple bits of each sample, with varying bit changes (1-4 bits) explored to optimize embedding. Results indicate that a 1-bit change in the LSB yields the best outcome, with minimal degradation of the host audio file. The method examines MSBs to determine the number of LSBs for data hiding, utilizing multiple and variable LSBs for embedding secret data. The process involves editing the LSB of data bytes, starting from a suitable position, and changing every alternate sample to embed the message.[2-5]

Again For example, if the word “Secret” has to be embedded into an audio file one has to embed the audio binary values of this word into the audio file. For this I have developed algorithm where multiple bits of each sample of the file have been changed or modified to insert text data in it. Also it is observed that the degradation of the host audio file after modification of the bits. The bit modification is done by different ways such as 1,2,3,4 bits were changed in

turn. But after checking through all modification bits, it has noticed that the best will get through 1-bit change in LSB. These methods check the MSBs of the samples, and then number of LSBs for data hiding is decided.[3]

In this way, multiple and variable LSBs are used for embedding secret data in host audio file. This process starts from a suitable position of the data bytes. Then the editing of the least significant bit of the data has been done that have to be embedded.

Taking every alternate sample and changing the least significant bit embeds the whole message.

II. Design Methodology

The initial model could be developed into an iterative model, with feedback from each phase influencing previous phases; this can be done by using “Waterfall Model”. Hence, we are implementing Waterfall Stages for Designing Steganography and cryptography are cousins in the spy craft family.

To enhance the initial model, an iterative approach can be adopted, incorporating feedback from each phase to refine previous stages, leveraging the Waterfall Model's structured methodology. This involves implementing Waterfall stages to design a robust steganography system.[6,8] Steganography and cryptography are complementary techniques used for secure communication. While cryptography encrypts messages to generate unreadable cipher text, steganography conceals the very existence of data within another medium. For instance, an encrypted text message may raise suspicions, whereas a steganographically hidden message remains undetectable. A ".wav" audio file is selected as the host, assuming minimal degradation of sound quality upon modifying its least significant bits. Unlike lossy, compressed MP3s, WAV files are lossless and uncompressed, ensuring superior audio quality. Regardless of bitrate, MP3s cannot match the fidelity of WAV files due to inherent compression losses.

WAV files have two basic parts, the header and data. Normally in wav files, the header is situated in the initial 44 bytes of file. Leaving the first 44 bytes, the rest of the bytes of the file are all about the data. The data is one chunk of samples that represents the whole audio. While embedding data, one can’t deal with the header sections because a minimal change in the header section leads to a corrupted audio file.

A program has been developed which can read the audio file and stores them in a different file. The first 44 bytes should be left unchanged because these are the data of the header section. Then the remaining data field is modified for embeding textual information. For example, if the word “Secret” has to be embedded into an audio file one has to embed the binary values of the word “Secret” into the audio data field[1].

Table 1. Letters with ASCII values and corresponding binary values.

Letters	Ascii values	Binary values
S	82	1010010
E	68	1000100
C	66	1000010
R	81	1010001
E	68	1000100
T	83	1010011

From the table, one can come to a point that for embedding the word "Secret" into the host audio file actually the corresponding eight bit binary values have to be embedded into the data field of that audio file.[4]

III. Design Algorithm

To develop this algorithm multiple bits of each sample of the file have been changed or modified to insert text data in it. It has also been observed the degradation of the host audio file after modification of the bits. The bit modification was done by various ways, like 1, 2, 3, 4 bits were changed in turn. But after going through all the modification it has been observed that 1-bit change in LSB gave the best result. Thus, data can be embedded according to the following algorithm. [4]

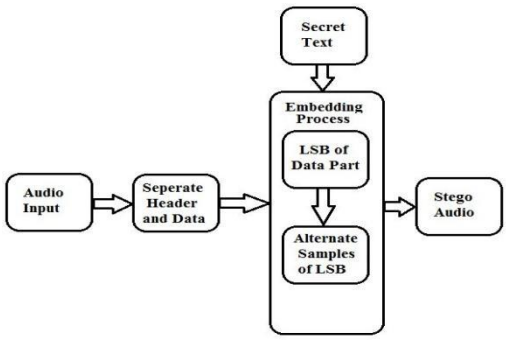

Fig 1. Algorithm (For Embedding of Data).

First audio input is to be given then the header and data of the host audio file are to be separated. Leaving the header part (that is first 50 bytes) untouched; the embedding process is done by taking every alternate sample and changing the Least Significant bit in a queue to embed the whole secret message.

The data retrieving algorithm at the receiver's end follows the same logic as the embedding algorithm with slight change. Algorithm (For Extracting of Data):

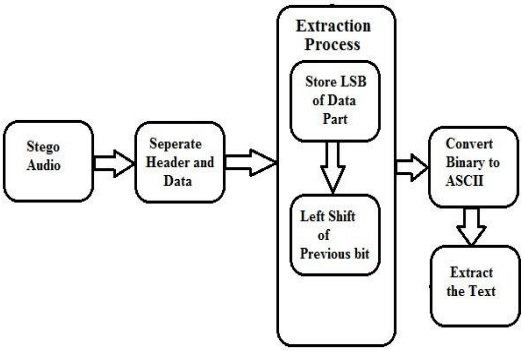

Fig 2. Algorithm (For Extracting of Data).

After the completion of the embedding process, stegno audio is obtained as the output. Now this Stegno output is the input for the extraction process. Leaving the first 50 bytes and starting from the 51st byte and store the least significant bit in a queue.

Check every alternate sample and store the least significant bit (LSB) in the previous queue with a left shift of previous bit. Convert the binary values to corresponding decimal to get the ASCII values of the secret message. From the ASCII the secret message can be found.

IV. Interpretation And Result

Sample No. Binary Values of corresponding sample Binary value to be embedded Binary values after Modification.

An audio file named “secret.wav” has been selected. After checking the binary values of each sample, first

50 samples were left without any changes. If the binary value of the corresponding sample is “01010010” then “0” should be modified. From Table I it can be observed that to embed the letter “S”, the sender has to embed the binary value “01010010”.

That is why according to the embedding algorithm “S” should be embedded according to Table II.

Table 2. Sample of Audio Files with Binary Before and After Embedding.

Sampl e No.	Binary Values of correspondin g sample	Binary value to be embedde d	Binary values after Modificatio n
51	01110100	0	01110100
53	01011110	1	01011111
55	10001011	0	10001010
57	01111011	1	01111011
59	10100010	0	10100010

61	00110010	0	00110010
63	11101110	1	11101111
65	01011100	0	01011100

Sample No. Binary value with embedded secret data Bits that are stored in Queue.

From Table II the embedding process of the letter "S" was discussed hence, in Table III, the extraction process of "S" is depicted. We have started from the 51st sample, every alternate sample is checked and the least significant bit is stored into a queue with a left shift of previous bit.

After getting all the bits in the queue, start from the left hand side, take 8 bits and convert them into corresponding decimal to get the ASCII, from the ASCII retrieve the embedded textual message. From the table, it is clearly observed that after getting 01010010 in the queue it is converted into the corresponding decimal that is 82, the ASCII of "S". Thus "S" is retrieved. Similarly, the next letters are also extracted and hence the complete word "SECRET'.

The below figure shows the graph of time and frequency domain plot of original audio file and encoded audio file. It shows graph remains same for both before and after embedding text , the overall size of the audio file remains the same. Thus, the undetectable or non-audible differences between the original and the stego audio files.

Table3: Frequency Domain Plot of Encoded Audio.

Sample No.	Binary value with embedded secret data	Bits that are stored in Queue
51	01110100	0
53	01011111	01
55	10001010	010
57	01111011	0101
59	10100010	01010
61	00110010	010100
63	11101111	0101001
65	01011100	01010010

which point it leads to perceptible change in the audible sound quality to any other third party other than the sender or receiver.

V. Application

The host signal should be non-objectionably degraded and the embedded data should be minimally

perceptible. The embedded data should be directly encoded into the media, rather than into a header or wrapper, so that the data remain intact across varying data file formats.

Error correction coding should be used to ensure data integrity. The embedded data should be self-clocking or arbitrarily re-entrant. This ensures that the embedded data can be recovered when only fragments of the host signal are available.

Military purpose At this time of competition between countries, there are many things which are needed to be kept secret or hide from the third party, for this purpose there is a novel approach to audio steganography in which embedding is done. For example, if the military of two countries wants to communicate between each other secretly and the existence of secret message have to be unknown to the third country, so they can use this stegnography method for encoding and decoding purpose.

This secure conversation can be saved from wrong hands. An application for data hiding is tamper-proofing. It is used to indicate that the host signal has been modified from its authored state. Modification to the embedded data indicates that the host signal has been changed in some way.

Another application, feature location, requires more data to be embedded. In this application, the embedded data are hidden in specific locations within an image. It enables one to identify individual content features, e.g., the name of the person on the left versus the right side of an image. Typically, feature location data are not subject to intentional removal

VI. Conclusion

We conclude that, this paper proposes a technique for embedding text-based data into a host audio file through bit modification. A custom procedure is developed to edit the data field, inserting intended information into the audio file. To ensure integrity, the header section is thoroughly examined, as even minor alterations can corrupt the entire file. The algorithm leaves the first 50 bytes untouched, modifying every alternate sample starting from the 51st byte to conceal textual data. While the impact of varying bit field modifications on performance is not explored, a preliminary study assesses the degradation caused by altering specific bit fields in the host audio file, presented herein.

VII. References

[1] Alhassan S, Daabo MI, Armah GK (2022) Twin K- shuffle based audio steganography. Asian J Eng Appl Technol 11(1):1–4

[2] S Hemalatha; Ramathmika “A Robust MP3 Audio Steganography with Improved Capacity” Published in 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 10.1109/ICCCA49541.2020. 9250894

[3] Dingwei Tan; Yuliang Lu; Xuehu Yan; Xiaoping Wang, “A Simple Review of Audio Steganography” Published in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 10.1109/ITNEC.2019.8729476

[4] Priyanka Joshi,Prof Ashwini Bade,”Modification Techniques of Audio Steganography for Secure Communication” Published in International Journal of Science, Engineering and Technology, 2022

[5] P.N.S. Sailaja, B.Chinnakrao, K.Rajeshwari, G.V.Shridhar, "Secret Communication Through Audio For Defense Application", International Conference on Electrical and Electronics Engineering (ICEEE) - 9th Sept, 2012, Guntur- ISBN: 978-93-82208-21-1.

[6] Prof. Samir Kumar, Bandyopadhyay Barnali, Gupta Banik, "LSB Modification and Phase Encoding Technique of Audio Steganography Revisited", Vol. 1, Issue 4, June 2012.

[7] S.S. Divya, M. Ram Mohan Reddy, "Hiding text in Audio using multiple LSB Steganography and provide Security using Cryptography", Vol.1, 6th July, 2012, ISSN 2277-8616 68 IJSTR©2012.

[8] W. Bender d. Gruhl n. Morimotoa. Lu in their "Techniques for data hiding" IBM systems journal, vol 35, nos 3&4, 1996.

[9] Peter H. W. Wong, Oscar C. Au, Justy W. C. Wong in their "Data Hiding and Watermarking in JPEG Compressed Domain By DC Coefficient Modification", Department of Electrical and Electronic Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

[10] Budda Lavanya, Yangala Smruthi, Srinivasa Rao Elisala, "Data hiding in audio by using image Steganography technique" Volume 2, Issue 6, November – December 2013 ISSN 2278-6856.

[11] Ravi Saini, Rajkumar Yadav C.M.R.A., GP Sanghi Rohtak, "A New data hiding method using pixel position and logical and operation", et al international journal of computer and electronics research [volume 1, issue 1, June 2012] ISSN: 2778-5795.

[12] Samir Kumar Bandyopadhyay, Indra Kanta Maitra "An Alternative Approach of Steganography using Reference Image", International Journal of Advancements in Technology, ISSN 0976-4860 Vol 1, No 1 (June 2010)