International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

Embedding Data in Audio and Image Processing by Steganographic Tool Techniques

Mrs. K.Shanthi,
M.Tech Assistant Professor,
Department of ECE,
Baba Institute of Technology and Sciences,
Visakhapatnam, India

Mr. K. Abhishek,
B.Tech, Final Year
Department of ECE,
Baba Institute of Technology and Sciences,
Visakhapatnam, India

***Abstract*-In modern communication system, data hiding is most essential for Network Security issue. Steganography means the ability of hiding the information in all the way. Itis an art of sending hidden data or secret messages over a public channel so that a third party cannot detect the presence of the secret messages.The goal of steganography is different from classical encryption, which seeks to conceal the content of secret messages In this paper we mainly discuss combination of steganographic methods i.e., method of embedding textual information in an audio file andmethod of embedding text or image in a image fileis presented in this paper**

***Keywords*— lsb, wav,encryption,decryption**

I. INTRODUCTION

Currently the fast improvement of the Internet and the digital information revolution caused major changes in the overall culture. The word steganography comes from the Greek Stegano˝s, which means covered or secret and graphy means writing or drawing. Steganography is the science of hiding secret information in a cover file such that only sender and receiver can detect the existence of the secret information. When hiding information inside images the LSB (Least Significant Byte) method is usually used. When hiding information inside Audio files the technique usually used is low bit encoding which is somewhat similar to LSB that is generally used in Images. During World War 2 invisible ink was used to write information on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as milk, vinegar and fruit juices were used, because when each one of these substances are heated they darken and become visible to the human eye. But it is not a perfect method to hide the data for sharing the information. So steganography was introduced the main goal of this paper was to find a way so that an audio file can be used as a host media to hide text. Using combination of image and audio stegnography. Because Steganography, in general, depends on the imperfection of the human auditory and visual systems

II.IMPLEMENTATION OF LSB CODING

A.AUDIO STEGANOGRAPHY

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. 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**.**

B.LSB CODING

In the current endeavor, an audio file with ".wav" extension has been selected as host file. It is assumed that the least significant bits of that file should be modified without degrading the sound quality.To do that, first one needs to know the file structure of the audio file. Like most files, WAV files have two basic parts, the header and the data. In normal wav files, the header is situated in the first 44 bytes of the file. Except the first 44 bytes, the rest of the bytes of the file are all about the data. The data is just one giant chunk of samples that represents the whole audio. While embedding data, one can't deal with the header section. That is because a minimal change in the header section leads to a corrupted audio file.

TABLE I

Letters with ASCII Values and Corresponding Binary Values

C.STEPS (FOR EMBEDDING OF DATA):

- Leave the header section of the audio file untouched.
- Start from a suitable position of the data bytes. (For the experiment purpose the present start byte was the 45th byte). Edit the least significant bit with the data that have to be embedded.
- Take every alternate sample and change the least significant bit to embed the whole message. The data retrieving algorithm at the receiver's end follows the same logic as the embedding algorithm.

D.STEPS (FOR EXTRACTING OF DATA):

- Leave first 44 bytes.
- Start from the 46th byte and store the least significant bit ina queue.
- Check every alternate sample and store the least significant bit in the previous queue with a left shift of the previous bit.
- Convert the binary values to decimal to get the ASCII values of the secret message.
- From the ASCII find the secret message.
- An audio file named "audio.wav" has been selected for this experiment. After checking the binary values of each sample, first 44 samples were left without any changes. The data embedding with LSB modification has been started after the header section. If the data embedding process is started from 45th sample then the LSB value of the 45th sample should be modified. If the binary value of the corresponding sample is "01110100" then "1" should be modified. From Table I it can be observed that toembed the letter "A", the sender has to embed the binary value "01000001". That is why according to the embedding algorithm "A" should be embedded according to Table II. Start from 46th bit, check the least significant bit, and store it in a queue. Check every alternate sample to collect the whole messages. Like 48th, 50th and 52th and so on. Store the least significant bits of the alternate samples in the queue with left shift of previous bit. Convert the binary values to decimal to get back the ASCII from which the text can be retrieved. The whole retrieval process can be depicted with the following table more thoroughly.

TABLE II

Samples of Audio File with Binary Values before and after Embedding

Sample No	Binary values of corresponding sample	Binary value to be embedded	Binary values after modification
46	01110100	0	01110100
48	01011110	1	01011111
50	10001011	0	10001010
52	01111011	0	01111010
54	10100010	0	10100010
56	00110010	0	00110010
58	11101110	0	11101110
60	01011100	1	01011101

TABLE III

Letter	ASCII Value	Corresponding Binary Value
A	065	01000001
U	117	01110101
D	100	01100100
I	105	01101001
O	111	01101111

Extraction of Data from Audio File

Sample No	Binary values with embedded secret data	Bits that are stored in the Queue
46	01110100	0
48	01011111	01
50	10001010	010
52	01111010	0100
54	10100010	01000
56	00110010	010000
58	11101110	0100000
60	01011101	01000001

As in Table II the embedding process of the letter "A" was stated that is why, in Table III, the retrieval process of "A" is depicted. Starting from the 46th sample, every alternate sample has been checked and the least significant bit has been 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 equivalent decimal to get the ASCII,from the ASCII retrieve the embedded textual message. From the table, it is clearly observed that aftergetting 01000001 in the queue it is converted into the equivalent decimal that is 60, the ASCII of "A". Thus "A" is retrieved. Like the same way, the next letters also have been retrieved and hence the complete word "Audio."

III.IMAGE STEGANOGRAPHY

In order to hide a message within a digital message, we take advantage of the least significant bit within each pixel of an image. Each pixel is made up of three eight bit integers that store the value of the color in each image. For example, 255 red, 255, green, and 0 blue makes the color yellow. By replacing the least significant bit in each of these color values, it is possible to hide a secret message, bit by bit, without changing the color values too much, as illustrated in Example 1. (Unaltered least significant bit. Encodedleast significant bit.)

TABLE IV

Variants	Red pixel Value	Green pixel value	Blue pixel value
Unaltered	1111 1111	1111 1111	0000 0000
Encoded	1111 1111	1111 1110	0000 0001
Message	1	0	1

Example 1: To hide the three bit message '101' in a pixel of the color yellow.

A.STEGANOGRAPHIC ENCODING PROCESS

Encoding hidden messages using steganography follows the basic process outline in Figure 1.

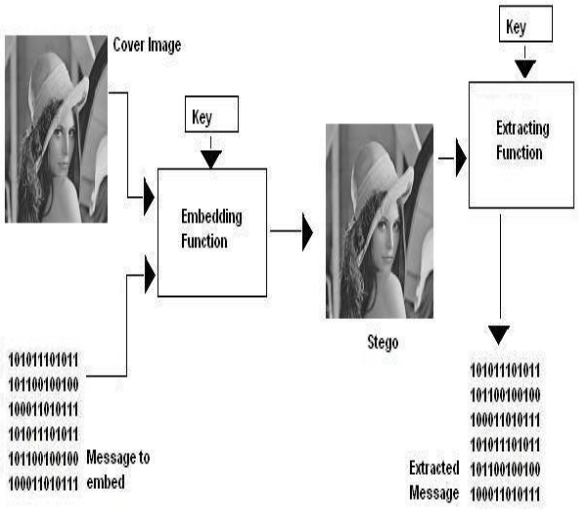


Figure1.Message encryption & decryption

The message and header are XOR encrypted and then encoded into the cover medium image. The message is first analyzed to determine whether it is a text or image message type. The message type and dimensions of the message (overall length for text and height/width values for an image) form an 8-bit Header that is used to reconstruct the message during the Decoding Process..

The Header is concatenated to the beginning of the message and this new combined message is encrypted using a simple symmetric Exclusive OR (XOR) encryption key, which follows the bit logic outlined in Table V and Example 2.

TABLE V

Message Value	EncryptionKey Value	Encrypted value
0	0	0
0	1	1
1	0	1
1	1	0

Table VI and Example 2: XOR Encryption Bit Logic. XOR provides a powerful tool for symmetric encryption because the same encryption key can be used to recover the Plaintext from Ciphertext.From Example 2 we can clearly see that we need to use the same encryption key during the decryption process if we want to successfully recover the original message. Unfortunately, this symmetric encryption key is a shared secret that must be sent separately from the encrypted message to ensure it remains uncompromised and secure.

TABLE VI

Encoding		**Decoding**	
Plain text	10101010	Cipher text	10100101
Encryption Key	00001111	Encryption Key	0000 1111
Cipher text	10100101	Recovered plain text	10101010

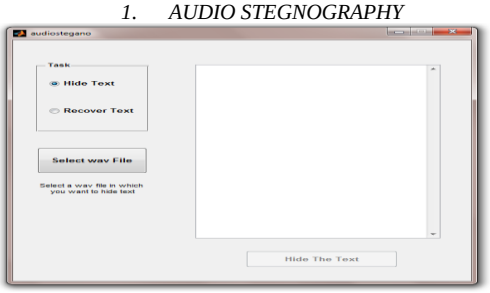
Finally, the encrypted Header and Message is encoded onto the Cover Medium Image's least significant bits using one of two methods. The first is a simple Sequential Encoding method and the second uses a Pseudo-Random Encoding method, which we will describe in more detail in the Method Section below

B.STEGANOGRAPHIC DECODING PROCESS

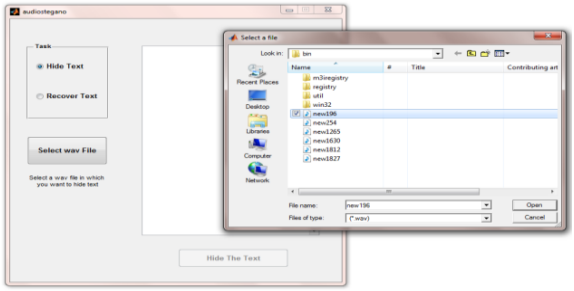
Decoding and recovering the hidden message follows the basic process outlined in Figure 1. This process reverses the effects of the encoding process and reveals the secret message to an authorized user. First, the Header is decoded, XOR decrypted, and analyzed to determine the dimensions and message type. Next, the Message is decoded and XOR decrypted before finally being reconstituted using the dimension data from the Header. First the encrypted Header values are recovered from the Cover Medium Image and decrypted using the sameXOR encryption key used during the encoding process. TheHeaderdimension value (length for text message; width times height for image message) is used to determine the stop value for the recovery algorithm, thereby reducing the complexity and speeding up the recovery process.Next, the program recovers the entire encrypted Message from the Cover Medium Image by using the Header Dimensions to determine when to stop. The recovered Message is decrypted using the same XOR encryption key used during encoding. Finally, if the message type is an image, the program uses the height/width dimensions from the recovered Header to reconstruct the original image from the decrypted message values for display.Drop down menu to differentiate the head from thetext.Some of the applications are Enables secret communication, Data hiding in audio or video is of interest for the protection of copyrighted digital media,tremendous use in Military Applications,in forensic applications for inserting hidden data into audio files for the authentication of spoken words and other sounds.

SIMULATED RESULTS IN MATLAB

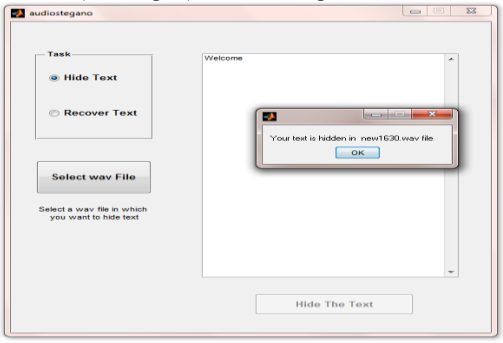
1. AUDIO STEGNOGRAPHY



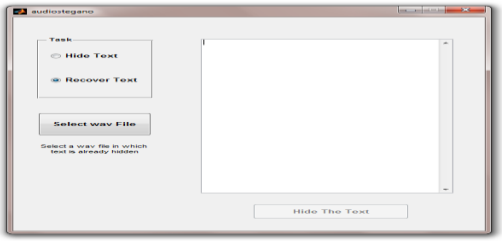
1. Guide: Graphical User Interface Tools.



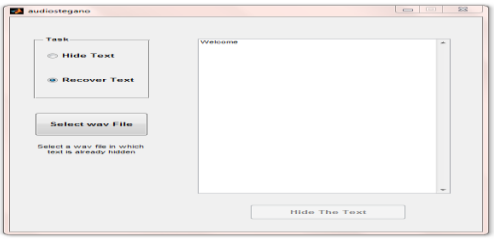
2. Select Wav file (cover signal) for Embedding



3. Encryption



4.Initialize Recover Process



5.DecryptionProcess

2.IMAGE STEGNOGRAPHY

Figure 2

Figure 3

CONCLUSION

In this paper we implemented relatively simple Sequential and Pseudo-Random Encoding and Decoding techniques using MATLAB for the functions. Converting these functions into a more efficient and web-friendly format is a natural extension of this work and would provide an interesting comparison.Even though steganography makes it difficult to detect the presence of a hidden message, steganalysis uses statistical analysis tools and processes to identify and recover message data from a cover image and Audio Steganography algorithms namely LSB Coding and implemented the same using MATLAB.At the end, feasibility of Audio Steganography was evaluated by considering it's the pros and cons

REFERENCES

1. N. Provos, P. Honeyman, *Hide and Seek: An Introduction to Steganography*, IEEE Computer Security 2003, <http://www.citi.umich.edu/u/provos/papers/practical.pdf>.
2. Wikipedia, *Microdot*, http://en.wikipedia.org/wiki/Microdot
3. S. Singh, The Code Book, Anchor Books, 2000, ISBN: 0385495323.
4. Data Hiding by LSB Substitution Using GeneticOptimal Key-Permutation
5. Pramatanathbasu&tanmayBhowmik 2010.International conference on recent trends in information,telecommunication and computing ,on Embedding of Text in Audio-Acase of Stegnagraphy
6. Johnson Neil F.andStefanKatzenbeisser."A Survey of StegnographicTechniques",In Information Hiding:Techniques for steganography and Digital watermarking.Boston ,Artech House.43-78.2000
7. Mohammad pooyan Ahmed Delforouzi,'LSB-based Audio Stegnanography method based on lifting wavelet Transform."International Symposium on Signal Processing and Information Technology,IEEE,2007

,