Audio steganography using LSB encoding technique with increased capacity and bit error rate optimization

Conference Paper · October 2012

DOI: 10.1145/2393216.2393279

CITATIONS
5

READS
1,752

4 authors, including:

Sangita Roy
Indian Institute of Technology Patna
14 PUBLICATIONS **53** CITATIONS

SEE PROFILE

Avinash Kumar Singh
Indian Institute of Information Technology Allahabad
21 PUBLICATIONS **69** CITATIONS

SEE PROFILE

Ashok Sairam
Indian Institute of Technology Guwahati
56 PUBLICATIONS **120** CITATIONS

SEE PROFILE

Audio Steganography Using LSB Encoding Technique with Increased Capacity and Bit Error Rate Optimization

Sangita Roy
Indian Institute of Technology, Patna
Bihar
India
+91-8521309190
r_sangita@iitp.ac.in

Avinash Kumar Singh
Indian Institute of Information Technology, Allahabad
Allahabad
India
+91-9807039372
rs110@iiita.ac.in

Jyotirmayee Parida
Modern Institute of Technology and Management
Bhubaneswar
India
+91-9861227642
jyotirmayeeparida@gmail.com

Ashok Singh Sairam
Indian Institute of Technology, Patna
Bihar
India
+91-8521309190
ashok@iitp.ac.in

ABSTRACT
Staganography is the art and science of secret hiding. The secret message or plain text may be hidden in one various ways. The methods of cryptography render the message unintelligible to the outsider by various transformations of the text whereas the methods of steganography conceal the existence of the message. To conceal a secret message we need a wrapper or container as a host file. Different wrappers or host files or cover medium are used to hide the secret message e.g. image, audio, video, text. The work in this paper aims at enhancing the provision of audio steganography by introducing one LSB (Least Significant Bit) coding technique. We design a high bit rate LSB audio watermarking method that reduces embedding distortion of the host audio with increased capacity of secret text. By using standard and proposed algorithm, watermark bits are embedded into higher LSB layer, resulting in increased robustness against noise addition, which is limited by perceptual transparency.

Categories and Subject Descriptors
C.2.0 [**Security and Protection**]: Language Contructs and Features – *Cryptography, Staganography, Encryption, Decryption.*

General Terms
Security

Keywords
Steganography, LSB encoding,

CCSEIT-12, October 26-28, 2012, Coimbatore [Tamil nadu, India]

1. INTRODUCTION
The scientific study in the open literature began in 1983 when Simmons stated the problem in terms of communication in a prison [3]. In his formulation, two inmates Alice and Bob are trying to hatch an escape plan. The only way they can communicate with each other is through a public channel, which is carefully monitored by warden of the prison ward. If ward detects any encrypted messages or code, he will throw both Alice and Bob into solitary confinement. The problem of steganography is introduced then; how can Alice and Bob cook up an escape plan by communicating over the public channel in such a way that Ward doesn't suspect "anything unusual" is going on. Notice, how the goal of steganography is different from classical cryptography, which is about hiding the content of secret message: steganography is about hiding the very existence of the secret message [7].

Steganographic protocols have a long and intriguing history that goes back to antiquity. There are stories of secret messages written in invisible ink or hidden in love letters (the first character of each sentence can be used to spell a secret, for instance). More recently, steganography was used by prisoners and soldiers during World War II because all mails in Europe was carefully inspected at the time [1]. Postal censors crossed out anything that looked like sensitive information (e.g. long strings of digits), and they prosecuted individuals whose mail seemed suspicious. In many cases, censors even randomly deleted innocent-looking sentences or entire paragraphs in order to prevent secret messages from going through. Over the last few years, steganography has been studied in the framework of computer science, and several algorithms have been developed to hide secret messages in innocent looking data [10].

2. LSB ENCODING
Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file [9]. 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

following diagram illustrates how the message 'HEY' is encoded in a 16-bit CD quality sample using the LSB method:

In LSB coding, 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. This increases the amount of data that can be encoded but also increases the amount of resulting noise in the audio file as well. Thus, 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 LSB coding until the message has been completely embedded, leaving the remaining samples unchanged. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples. Yet now the embedding process ends up changing far more samples than the transmission of the secret required. This increases the probability that a would-be attacker will suspect secret communication.

A more sophisticated approach is to use a pseudorandom number generator to spread the message over the sound file in a random manner. One popular approach is to use the random interval method, in which a secret key possessed by the sender is used as a seed in a pseudorandom number generator to create a random sequence of sample indices. The receiver also has access to the secret key and knowledge of the pseudorandom number generator, allowing the random sequence of sample indices to be reconstructed. Checks must be put in place, however, to prevent the pseudorandom number generator from generating the same sample index twice. If this happened, a collision would occur where a sample already modified with part of the message is modified again. The problem of collisions can be overcome by keeping track of all the samples that have already been used. Another approach is to calculate the subset of samples via a pseudorandom permutation of the entire set through the use of a secure hash function. This technique insures that the same index is never generated more than once.

2.1 Standard LSB

Data hiding in the least significant bits (LSBs) of audio samples in the time domain is one of the simplest algorithms with very high data rate of additional information [2]. The LSB watermark encoder usually selects a subset of all available host audio samples chosen by a secret key. The substitution operation on the LSBs is performed on this subset, where the bits to be hidden substitute the original bit values. Extraction process simply retrieves the watermark by reading the value of these bits from the audio stego object. Therefore, the decoder needs all the samples of the stego audio that were used during the embedding process. The random selection of the samples used for embedding introduces low power additive white Gaussian noise (AWGN). It is well known from the psychoacoustics literature that the human auditory system (HAS) is highly sensitive to AWGN. That fact limits the number of LSBs that can be imperceptibly modified during watermark embedding.

The main advantage of the LSB coding method is a very high watermark channel bit rate; use of only one LSB of the host audio sample gives capacity of 44.1 kbps (sampling rate 44 kHz, all samples used for data hiding) and a low computational complexity. The obvious disadvantage is considerably low robustness, due to fact that simple random changes of the LSBs destroy the coded watermark.

As the number of used LSBs during LSB coding increases or, equivalently, depth of the modified LSB layer becomes larger, probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. Therefore, there is a limit for the depth of the used LSB layer in each sample of host audio that can be used for data hiding. Subjective listening test showed that, in average, the maximum LSB depth that can be used for LSB based watermarking without causing noticeable perceptual distortion is the fourth LSB layer when 16 bits per sample audio sequences are used. The tests were performed with a large collection of audio samples and individuals with different background and musical experience. None of the tested audio sequences had perceptual artifacts when the fourth LSB has been used for data hiding, although in certain music styles, the limit is even higher than the fourth LSB layer. Robustness of the watermark, embedded using the LSB coding method, increases with increase of the LSB depth used for data hiding. Therefore, improvement of watermark robustness obtained by increase of depth of the used LSB layer is limited by perceptual transparency bound, which is the fourth LSB layer for the standard LSB coding algorithm.

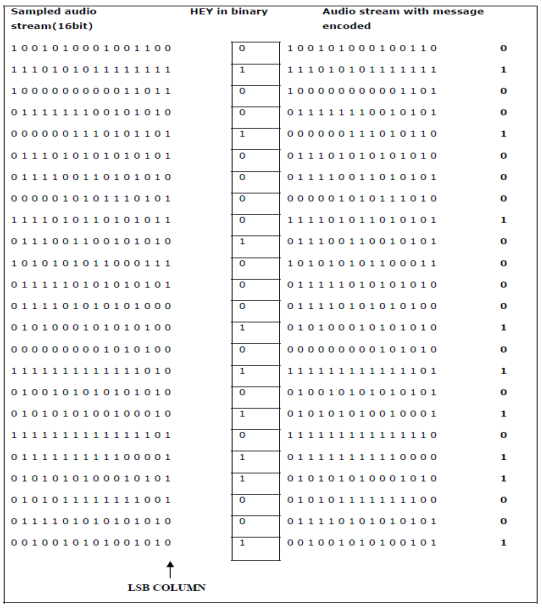

Figure 1. LSB encoding technique.

2.2 Modified LSB

This method is able to shift the limit for transparent data hiding in audio from the fourth LSB layer to the sixth LSB layer, using a two- step approach [4]. In the first step, a watermark bit is embedded into the ith LSB layer of the host audio using a LSB coding method. In the second step, the impulse noise caused by watermark embedding is shaped in order to change its white noise properties. The standard LSB coding method simply replaces the original host audio bit in the ith layer (i=1,...,16) with the bit from the watermark bit stream. In the case when the original and watermark bit are different and ith LSB layer is used for embedding the error caused by watermarking is 2i[1] quantization steps (QS)(amplitude range is [-32768, 32767]). The embedding error is positive if the original bit was 0 and watermark bit is 1 and vice versa. The key idea of the proposed LSB algorithm is watermark bit embedding that causes minimal embedding distortion of the host audio. It is clear that, if only one of 16 bits in a sample is fixed and equal to the watermark bit, the other bits can be flipped in order to minimize the embedding error. For example, if the original sample value was $(0...01000)_2=(8)_{10}$, and the watermark bit is zero is to be embedded into 4^{th} LSB layer, instead of value $(0...00000)_2=(0)_{10}$, that would the standard algorithm produce, the proposed algorithm produces sample that has value $(0...00111)_2=(7)_{10}$, which is far more closer to the original one. However, the extraction algorithm remains the same; it simply retrieves the watermark bit by reading the bit value from the predefined LSB layer in the watermarked audio sample. In the embedding algorithm, the (i+1)th LSB layer (bit a*i*) is first modified by insertion of the present message bit. Then, the algorithm given below is run. In case that the bit a*i* need not be modified at all due to being already at a correct value, no action is taken with that signal sample[16].

To hide a message in wave sample grab one carrier unit, put one bit of the message into the lowest 4^{th} bit of the carrier unit, flip the rest one and write the changed unit to the destination stream.

LSB coding is explained in the following procedure:

1. Read one sample from the wave stream.
2. Get the next bit from the current message byte.
3. Place it in the current 4th bit of the sample.
4. Flip the rest 3 bits accordingly.
5. Copy the rest of the wave without changes.

3. PROPOSED LSB TECHNIQUE WITH INCREASED CAPACITY

3.1 Technique and Algorithm

In the modified LSB encoding technique we have seen that the 4th bit is set according to the secret message. If the sample bit is not equal to the secret message bit we then simply flip the rest of the bits of that given sample. In our proposed model we take consecutive two bits from the secret message and instead of changing a single bit in a sample we change two bits (4th and 3rd position) of the sample. If there is change in this two bits we flip rest of the LSB otherwise there is no change. For example, if the original sample value was $(0...01000)_2=(8)_{10}$, and the watermark bits 01 are to be embedded into 4th and 3rd LSB layer, the standard algorithm will produce the value $(0...00000)_2=(0)_{10}$ to embed the 1st watermark bit only and for the 2nd bit we need another sample, the modified algorithm produces sample that has value $(0...00111)_2=(7)_{10}$, which is far more closer to the original one but here also we need another sample to embed the 2^{nd} watermark bit. Our proposed algorithm will produce $(0...00111)_2 = (7)_{10}$ which is equal to the value produced by modified LSB technique but this sample contains two watermark bit (here 0 and 1) instead of one bit. So we can say that with the same bit error we increased the capacity of the sample to hide more secret message.

Our proposed LSB technique is explained in the following procedure:

1. Read one sample from the wave stream.
2. Get the next two bits from the current message byte.
3. Place it in the current 4th and 3rd bit of the sample.
4. Flip the rest 2 bits accordingly.
5. Copy the rest of the wave without

For example, suppose we have 4 samples to hide a secret message 01010101

Table 1. Bit flipping using two different algorithm

Sample value	Two bits to embed	After embedding	One bit to embed	After Embedding
$(0..0100)_2$	01	$(0..0100)_2$	0	$(0..0100)_2$
$(0..0101)_2$	01	$(0..0101)_2$	1	$(0..1010)_2$
$(0..0110)_2$	01	$(0..0101)_2$	0	$(0..0110)_2$
$(0..0111)_2$	01	$(0..0101)_2$	1	$(0..1000)_2$

It is clear that the proposed method introduces smaller error and higher capacity during water- mark embedding. The secret message is completely embedded when we are trying two embed two bit values but when we are embedding 1 bit value then last 4 bits are not getting any sample to be inserted. If the 4th LSB layer is used, the absolute error value ranges from 1 to 4 QS, while the standard method in the same conditions causes constant absolute error of 8 QS. The average power of introduced noise is therefore 9.31 dB smaller if the proposed LSB coding method is used. In addition to decreasing objective quality measure, expressed as signal to noise ratio (SNR) value, proposed method introduces, in the second step of embedding, noise shaping in order to increase perceptual transparency of the method. A similar concept, called error diffusion method is commonly used in conversion of true color images to palette based color images. In our algorithm, embedding error is spread to the four consecutive samples, as samples that are predecessors of the current sample cannot be altered because information bits have already been embedded into their LSBs. Let e(n) denote the embedding error of the sample a(n). For the case of embedding into the 4th LSB layer, the next four consecutive samples of the host audio are modified according to these expressions:

$$a(n+1)=a(n+1)+\lfloor e(n)\rfloor \quad a(n+3)=a(n+3)+\lfloor e(n)/3\rfloor$$
$$a(n+2)=a(n+2)+\lfloor e(n)/2\rfloor \quad a(n+4)=a(n+4)+\lfloor e(n)/4\rfloor$$

where $\lfloor A \rfloor$ denotes floor operation that rounds A to the nearest integer less than or equal to A. Error diffusion shapes input impulse noise, introduced by LSB embedding, by smearing it and changing its distribution to a perceptually better-tuned one. Effect

is most emphasized during silent periods of audio signal and in fragments with low dynamics e.g. broad minimums or maximums. The both embedding steps jointly increase the subjective quality of audio stego object. Therefore, we expect that, using the proposed two-step algorithm, we can increase the depth of watermark embedding further than the 4th LSB layer and accordingly increase algorithm's robustness towards noise addition.

4. EXPERIMENTAL DETAILS

4.1 Digital Audio Processing

Matlab supports multi-channel wave format with up to 16 bits per sample [5][8]. To load a wave file, you can use "[Y, Fs, Nbits] = wavread (wave_filename)", where wave_filename is the file name of the wave file, Y the sampled data with dimension number of samples number of channels, Fs (in Hz) the sampling rate, and Nbits the number of bits per sample used to encode the data in the file. Amplitude values in Y vector are normalized to the range [-1, +1] according to the formula, Y = X /[2(Nbits-1)] – 1, where X is the original unsigned Nbits integer expression. For instance, while X is 128 with Nbits = 8, Y is 0. To generate a wave file and store it in the hard disk,you can use "wavwrite(Y, Fs, Nbits, wave_filename)". To playbackthe signal in vector Y, use "sound(Y, Fs, Nbits)".

4.2 Read, Playback and Visualize and Audio Signal

We use a few examples related to audio and image processing as warm-up exercises before we get into the design lab section. Let's start Matlab and type "Edit" in the command window. You will see a "M-file Editor". We will use this editor3 to write M-files[6].
Read, Playback and Visualize an Audio Signal
(1) Download the "symphonic.wav" audio file from the course web site. Make sure it is in your working directory.
(2)You can read an audio file into an Y matrix using the function wavread:
->[Music,Fs, Nbits] = wavread('symphonic.wav');
(3) To obtain the dimension of this image, type
-> [MusicLength, NumChannel] = size(Music);
The function size will return the number of samples and number of channels of this audio file into MusicLength and NumChannel.
(4) To playback this audio vector, type
->sound(Music, Fs, Nbits);

Make sure your speaker or earphone is on.
(5) We can visualize the waveform by typing:
->Display_start = 1;
-Display_end = MusicLength;
-> subplot(2,1,1); plot(Music(Display_start: Display_end,1));
-> title('First channel');
subplot(2,1,2); plot(Music(Display_start: Display_end,2));
->title('Second channel');
You can adjust the display range by changing Display_start and Display_end.

4.3 Bits Manipulation

(1) Convert the double value format of Music to an unsigned integer expression with Nbits
->IntMusic = (Music+1)*power(2,Nbits-1);
(2) Many music files use 16 bits to represent an audio sample. In this case, NBits =16. We can extract the lower byte from the first channel:
-> LowNbits = Nbits/2;
->LowIntMusicCh1 = zeros(MusicLength,1);
->FirstChannel = IntMusic(:,1); % extract the first channel
>for ibit = 1:1:LowNbits
->LowIntMusicCh1 = LowIntMusicCh1+ bitget(FirstChannel, ibit)*power(2,ibit-1);
->end
(3) Convert unsigned integer to the normalized expression and listen to it.
-> LowRecMusicCh1 = LowIntMusicCh1/power(2,LowNbits-1) - 1;
->sound(LowRecMusicCh1,Fs, LowNbits);
(4) Repeat the procedure for the second channel and store the final result in LowRecMusicCh2. What do you hear?

5. RESULTS AND ANALYSIS

Modified LSB watermarking algorithm was tested on 11 audio sequences from different music styles (pop, rock, techno, jazz). The audio excerpts were selected so that they represent a broad range of music genres, i.e. audio clips with different dynamic and spectral characteristics. All music pieces have been watermarked using the proposed and modified LSB watermarking algorithm. Clips were 44.1 kHz sampled mono audio files, represented by 16 bits per sample. Duration of the samples ranged from 10 to 15 seconds. As defined above, signal to noise ratio for the embedded watermark is computed as:

$$SNR = 10 \cdot \log_{10} \Sigma n\, x2(n) / \Sigma n\, [\, x2(n) - y2(n)]$$

Where x(n) represents a sample of input audio sequence and y(n) stands for a sample of audio with modified LSBs.

Table 2. Comparison study of 1 bit and 2 bit embedding

Wave file	Bit Rate	No of errors for 1 bit stego	No of errors for 2 bit stego	Bit error rate for 1 bit stego	Bit error rate for 2 bit stego
1	64	26	19	.0813	.0594
2	64	20	30	.0625	.0938
3	64	40	08	.1250	.0250
4	64	15	30	.0469	.0938
5	64	25	30	.0781	.0938
6	88	45	11	.1406	.0344
7	88	24	24	.0750	.0750
8	174	23	19	.0719	.0594
9	176	37	14	.1156	.0437
10	176	26	17	.0813	.0531
11	176	40	12	.1250	.0375

Results of subjective tests showed that the bit error rate is high when we are using 64 kbps sample audio for proposed LSB. But when we go for the higher kbps sample audio then the bit rate is lesser and it is minimized noticeably.

Therefore, a significant improvement in robustness against signal

processing manipulation can be obtained, as the hidden bits can be embedded two LSB layers deeper than in the standard LSB method. In order to compare the robustness of the proposed algorithm and the standard one, additive white Gaussian noise was added to the samples of watermarked audio and bit error rate (BER) measured.

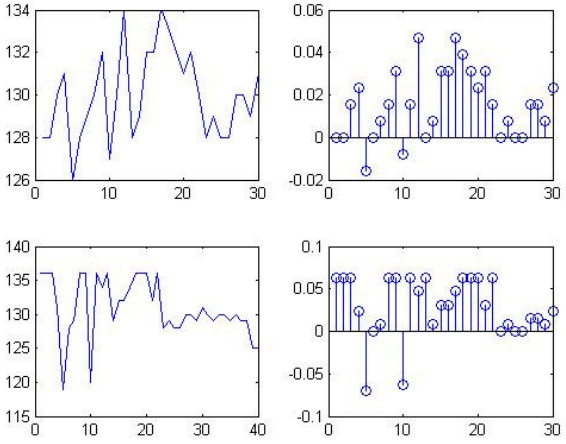

Figure 2. Before and after embedding secret message using 1bit embedding.

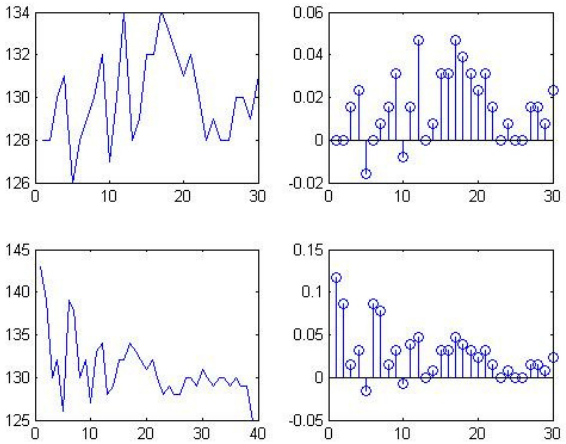

Figure 3. Before and after embedding secret message using 2bits embedding.

Upper portion of figure 2 and figure 3 depicts the sample audio signal before embedding data where the lower portion of both the figure shows the characteristics of signal after embedding the secret message.

6. CONCLUSION

We present an intelligent framework for steganography using LSB encoding for audio data. The idea of the algorithm is that two watermark bits are embedding which provides the minimal distortion of the fixed length host audio with high capacity. Listening test showed that described algorithm succeeds in taking two bit positions to embed secret data without affecting the perceptual transparency of the watermarked audio signal. The improvement in robustness in presence of additive noise is obvious, as the proposed algorithm obtains significantly lower bit error rates than the standard algorithm with an intelligent frame work, where the layer will be automatically chosen by the system and a single position is used to hide data, so we required more no of sample to hide the secret text.

Steganography has a number of drawbacks when compared to encryption. It required a lot of overhead to hide a relatively few bits of information. Once the system is discovered it becomes virtually worthless. This problem too can be overcome if the insertion method depends on some sort of key. Alternatively a message can be first encrypted and then hidden using steganography

7. REFERENCES

[1] Anderson, R, Bowman, Petticolas,F. On the limits of Steganography. IEEE Journal selected areas in Communication,16, 4, 474-481

[2] Bassia, P., Pitas I., Nikolaidis N. Robust audio watermarking in the time domain, IEEE Transactions on Multimedia 3, 2, 232-241.

[3] Brian, J., Yuliya K. and Andrew, L.Fröhlich.2006. audio Steganography Dec 13.

[4] Cedric, T., Adi, R.,Mcloughlin, I.: Data concealment in audio using a nonlinear frequency distribution of PRBS coded data and frequency domain LSB insertion, Proc. IEEE Region 10 International Conference on Electrical and Electronic Technology, Kuala Lumpur, Malaysia, 275-278.

[4] Cedric, T., Adi, R., Mcloughlin, I. Data concealment in audio and frequency domain LSB insertion, Proc. IEEE Region 10 International Conference on Electrical and Electronic Technology, Kuala Lumpur, Malaysia, 275-278.

[5] Fridrich, J., Goljan, M., Du, R.: 2002 Lossless Data Embedding – New Paradigm in Digital Watermarking, Applied Signal Processing, 2002, 2, 185-196.

[6] I. Cox, M. Miller and J. Bloom, 2003.Digital Watermarking Morgan Kaufmann Publishers, San Francisco, CA, 2003

[7] Krista, B., 2004. Linguistic steganography: survey, analysis and robustness concerns for hiding information in text, Center for Education and Research in Information Assurance and Security, Tech report 2004.

[8] Mobasseri, B. Direct sequence watermarking of digital video using m-frames Proc. International Conference on Image Processing, Chicago, IL, 399-403

[9] Roy, S., Manasmita M., 2011. A novel approach to format based test stegonography, International conference on communication computing and security, ICCCS 2011,Procidings by ACM with ISBN-978-1-4503- 0464-rourkela,Odisha,India.

[10] "Steganography FAQ" Aelphaeis Mangarae, march 18 2006..