ISSN: 0976-3376

Available Online at http://www.journalajst.com

ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Asian Journal of Science and Technology
Vol. 07, Issue, 07, pp.3216-3222, July, 2016

RESEARCH ARTICLE

A SECURE 2 LAYER LSB BASED AUDIO STEGANOGRAPHY USING HUFFMAN CODING

[1]Ratul Chowdhury,[*2]Samir Kumar Bandyopadhyay and [3]Anirban Das

[1]Department of Computer Science and Engineering, Future Institute of Engineering and Management, Kolkata, India
[2]Department of Computer Science and Engineering, University of Calcutta, Kolkata, India
[3]UG student, Jogesh Chandra Chowdhuri College, Kolkata, India

ARTICLE INFO

Article History:
Received 28th April, 2016
Received in revised form
21st May, 2016
Accepted 19th June, 2016
Published online 30th July, 2016

Key words:
Audio steganography;
Huffman coding;
Sparse matrix;
LSB coding.

ABSTRACT

Inthis paper we present a novel audio steganography method where the concept of Huffman coding and sparse matrixhas been introduced. The proposed method has reduced the secret message using Huffman coding and in next phase it exploits the sparse representation of the reduced string to embed secret message into higher semantic level of the cover audio file.For embedding purpose,the traditional LSB method has been used. The proposed method maintains both the stego signal quality and embedding capacity, which are the two major requirements of any steganography algorithm. The experimental resultillustrates that the stego signal generated by proposed method are perceptually indistinguishable from the original cover file. The experimental result is qualified by two quality measure Mean square errorand Signal to Noise Ratio. When compared with other methods, the proposed method is shown to be superior on addressing major requirements, imperceptibility, undetectability and capacity major.

INTRODUCTION

Information security is the technique of restricting information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. Increasing need of secure private information transmission over internet inspired by many researchers to make great efforts to explore steganography methods. The rapid e-communication over the internet has led to the use of three security techniques: cryptography, watermarking and steganography. In cryptography, the content of the messages are mangled. In watermarking, data are hidden to convey some information such as ownership and copyright. The third one is steganography. The word Steganography was derived from ancient Greek words steganos meaning "covered, concealed or protected" and graphein meaning "writing". Undetectebility is the major issue of any steganography algorithm. Undetectability ensures that the stego signal, containing the secretmessage is indistinguishable from the original signal bymeans of either the human auditory system (HAS) or steganalysismethods.

Corresponding author:Samir Kumar Bandyopadhyay,
Department of Computer Science and Engineering, University of Calcutta, Kolkata, India

Unlike cryptography, the message is unaltered in steganography. Here the messages are hidden within a media in such a way that none can understand the very existence of the message i.e. it cannot be perceived by human. Such systems only make sense when there is an intruder. This intruder may be passive and merely observe the traffic, or he may be active and modify it. So the arrack can be of two type's passive and active attack. In steganography applications, passive attack is considered while its goal is to detect whether steganography is being used or not and it does not disturb the communication before detecting the suspicious signals. Active attacks such as noise addition, MPEG audio coding, cropping, time shifting, filtering, resampling, requantization etc. The combination of cryptography and steganography increases the level of security. In first level the secret message is encrypted by different cryptography algorithm and in next level the encrypted message is embedded into the cover file. In this proposed method, we have used Huffman coding which not only reduce the size of the secret message into a desirable length, at the same time it provides a level of encryption. Huffman coding is a lossless data compression algorithm, the idea behind it is to assign variable length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding

characters present in it. According to this algorithm, the most frequent character gets the smallest code and the least frequent character gets the largest code. In next part we have applied the concept of sparse matrix into the reduced string. In sparse matrix representation, only the value of nonzero elements is stored in memory to reduce the wastage of memory. In our method a 4*4 temporary matrix has been constructed from the reduced target string where most of elements are zero. We have done the second level encryption by sending the location of the nonzero elements that is their row and column number. The second advantage is that we don't have to send the value of the nonzero elements because we are working with the binary sequence where nonzero means 1.The traditional LSB method has been used for embedding purpose. It is the simplest steganography technique to embed the bits of the target string into the least significant position of the cover file. Modulating the least significant bit does not result in human-perceptible difference because the amplitude of the change is very small.

Related work

Steganography is the technique of secure communication where the existence of the communication itself cannot be detected. On the otherhand, steganalysis is the art of detecting the secret communication. The medium which is used to carry message are called cover media which is basically image, audio or video file over public media. The concept of cryptography is about concealing the content of the message where the concept of steganography is concealing the existence of the secret messages. The ultimate goal is to embed the messages into the cover media in such a way that its existence is undetectable (Malik, Hafiz *et al*., 2015). In steganography system, it is basically assumed that there is a secret key agreement between sender and receiver. The secret key has been send through a secure channel. On the other hand, the stego signal is transmitted through public channel. In the receiving side, by using the secret key, the hidden message is extracted from the stego file. To hide the existence of the secret message into the stego file, cryptography and steganography are sometimes combined. There are basically two types of attack in the field of data hiding: passive and active attack. The goal of passive attack is to detect weather stegagraphy is being used or not. Itdoes not disturb the communication before detecting the suspicious signals. Active attacks such as noise addition, MPEG audio coding, cropping, time shifting, filtering, resampling, requantization, etc, (Ahani *et al*., 2015).

Steganography algorithm works in two domain frequency: domain and time domain. In frequency domain audio steganography methods are the ones in which various transform domains such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) are used to embed the secret message in the coefficients of the cover audio. Another data hiding technique is least significant bit modification it works in time domain. LSB methods typically achieve both high payload and low perceptibility. However, because the fact that the data are hidden in the least significant bit may be known, LSB methods are vulnerable to extraction by unauthorized parties, (Ahani *et al*., 2015). So cryptography and steganography are now combined. Audio steganography technique must satisfy three core conditions.

- **Capacity:** Capacity means the amount of secret information that can be embedded within a cover audio file.
- **Transparency**: Transparency evaluates how well a secret message is embedded into a cover audio so that its existence is undefined.
- **Robustness:** Robustness measure is the ability of secret message to withstand against attacks.

The popular audio steganography techniques are (Kiah *et al*., 2011 and Zamani *et al*., 2009).

- **Low bit encoding:** It is technique to store the data into the least significant bits of the cover file. It is a simple technique and by using this technique and by using this technique a huge volume of text can be embedded into a cover file (Banerjee *et al*., 2013).
- **Phase coding:** Phase coding works by substituting the phase of an initial audio segment with a reference phase, this phase actually represents the hidden data.
- **Spread spectrum technique:** It is a technique designed to encode any stream of data via spreading the encoded data across as much of the frequency spectrum as possible (Roshidi Din *et al*., 2006).
- **Echo hiding:** Echo technique embeds data into a cover audio signal by introducing an echo; the hidden data can be adjusted by two parameters, the first one is amplitude and the second is offset. These two parameters represent the magnitude and time delay of the embedded echo.

The current trend of audio steganography combines the concept of cryptography and steganography to perform a powerful encryption [8, 9]. Lots of cryptographic algorithms (symmetric and asymmetric key) are used to perform the first level encryption of the message. RSA, AES, DES. These are the popular algorithm which is used to encrypt the message first after that by using traditional audio steganography method the encrypted version of the message is embedded into a cover audio file. Lots of algorithms from different domain are now used with LSB based audio steganography for security purpose. One approach is modulo operator (Datta *et al*., 2015) where the first level encryption is done by using modulo operator and in second level by using standard LSB method the encrypted version of the text is embedded into the cover file. Genetic algorithm base approach is another example. Genetic algorithm is a technique for optimization and search. Generating population, creation of fitness function and mutation are the three major parts of genetic algorithm. By using these three parts the message is encrypted in a well manner (Bhowal *et al*., 2013). An alternative approach is to perform the first level encryption by using XOR operator. Another method is spectrum manipulation where the frequency of the transmitted signal is deliberately varied to perform better encryption. Zigzag LSB method is another approach, where the binary value of the secret message is inserted into the last bit of the audio in a zigzag fashion. On an average, only half of the bits are altered in the audio file. So there are no noticeable sound variations of the audio file before and after hiding the data. Multiple least significant bits modification is sometimes used to accommodate large volume

of data (Kiah, 2011). This gives an increased robustness against noise addition. Lots of different idea has been used in image steganography. Chaos Based Spread Spectrum Image Steganography (CSSIS) a novel scheme of (CSSIS) involving the use of chaotic encryption and chaotic modulation in spread spectrum image steganography (SSIS)has been proposed (Satish *et al*., 2004), 3d steganography is another approach based on a novel multilayered embedding scheme to hide secret messages in the vertices of 3D polygon models (Chao *et al*., 2009).

Detailed method for encryption

Steps to build Huffman Dictionary

The compression of the secret message by using Huffman coding has been done by using four steps:

- Construction of a private key.
- Probability calculation of each character present in the private key.
- Huffman tree construction.
- Conversion of the target string into code word from the Huffman tree.

Construction of private key

A private key has been constructed into the sending side. The private key is basically a pangram string which contains all the characters and special symbols. There is an agreement between sender and receiver is that the same private key should be present in the receiving side.

Probability calculation of each character present at the private key

Probability of occurrence of each character into the private key is calculated by (Total number of occurrence of each character)/ (the length of the private key). The calculate probability algorithm describes the exact method and its output shown in Table 1.

Algorithm calculate probability

Input: The pangram string

Output: Probability of each character present in the pangram string.

```
1. Start
2. len = length(data)
3. sym[1] := data[1]+0
4. prob[1] := 1;
5. i := 2
6. While i <= len :
7.         temp := data[i]+0
8.         flag := 1
9.         k := 1
10.        While k <= length(sym)
11.                If(sym[k] = temp) then
12.                        flag := 0
13.                EndIf
14.                k := k+1
15.        EndWhile
16. If(flag == 1)  then
17. sym[k+1] := temp
18. EndIf
19.  i := i+1
15.  EndWhile
16.  x := 1
17.  i := 1
18.  While i <= length(sym)
19.          count := 0
20.          k := 1
21.          While k <= len
22.                  If(sym[i] = data[k]+0) then
23.                          count:= count+1
24.                  EndIf
25.                  k := k+1
26.          EndWhile
27.          prob[x] := count/len
28.          x := x+1
29.          i := i+1
30.  EndWhile
31.  data[0] := prob
32.  data[1] := sym
33.  Stop
```

String:-"i love computer science."

Symbol	Probability
105	0.0833
32	0.1250
108	0.0417
111	0.0833
118	0.0417
101	0.1667
99	0.1250
109	0.0417
112	0.0417
117	0.0417
116	0.0417
114	0.0417
115	0.0417
110	0.0417
46	0.0417

Table 1. Probability calculation of each character Present in the private key

Huffman tree construction

In this phase the input is an array of unique characters with their frequency of occurrences and output is the Huffman Tree. The following steps illustrate the construction.

Algorithm Huffman_TreeConstruction

String:- "i am a programmer"

Probabilities:-

```
i       = 0.0588
(space) = 0.1765
a       = 0.1765
m       = 0.1765
p       = 0.0588
r       = 0.1765
o       = 0.0588
g       = 0.0588
e       = 0.0588
```

Huffman Data:-

```
i  = 00000
_  = 010
a  = 011
m  = 10
p  = 00001
r  = 11
o  = 0010
g  = 0011
e  = 0001
```

- Create a leaf node for every unique character and add it to the priority queue.
- Repeat Step 2.1 to 2.3 while no_of_nodes > 1[Here, no_of_nodes is number of nodes in the priority queue.]
  - Delete 2 nodes from the queue which has the lowest frequency (Highest priority).
  - Create a new node with the frequencies equal to the sum of the two node's frequencies. Then make the first taken node is the left child and the second node is the right child of new node.
  - Insert the new node in the queue.
- The remaining node of the queue is root node of the tree.

Figure 1 shows the tree construction

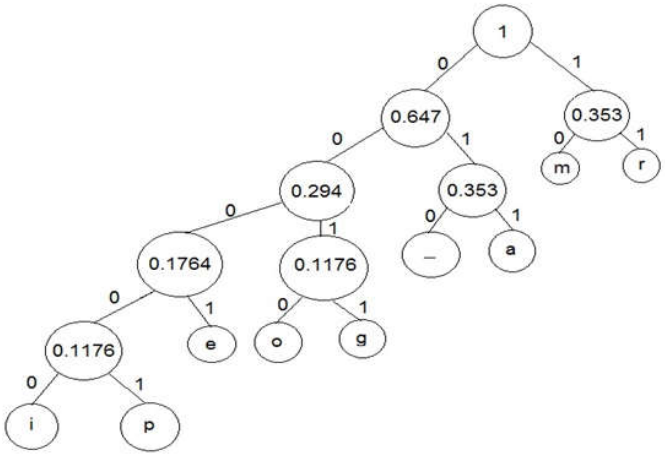

Figure 1. Huffman tree construction

Construction of temporary matrix from the reduced target string

The reduced string is nothing but a binary sequence and it should be a multiple of 16.If the sequence is not multiple of 16, padding some extra 0 at the beginning of the string has been done. A temporary 4*4 matrix has been constructed form the reduced target string shown in Figure 2.

$$M(1) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$M(2) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure2. Construction of temporary matrix

Row and column number estimation of each nonzero element

From each 4*4 temporary matrix we have to find the location number of each nonzero element. A temporary table has been constructed which contains the location of each nonzero elements and its corresponding binary equivalent. These binary equivalent values are further used for LSB replacement.The temporary table construction from matrix 1 and 2 are shown in Table 2 and 3.

Table 2. Number of nonzero elements with their location from Figure 2 first matrix

No of nonzero elements	Row value	Column value
1	0(00)	1(01)
2	0(00)	2(10)
3	1(01)	3(11)
4	2(10)	1(01)
5	2(10)	2(10)
6	3(11)	3(11)

Table 3.Number of nonzero elements with their location from Figure 2 second matrix

No of nonzero elements	Row value	Column value
1	0(00)	1(01)
2	0(00)	2(10)
3	1(01)	2(10)
4	1(01)	3(11)
5	2(10)	1(01)
6	2(10)	2(10)
7	3(11)	1(01)

Size estimation

Each temporary matrix contains 16 characters of the target string. So the number of temporary matrix from the target string will be :- (No of characters in the target string/16).If the target string is not a multiple of 16,add some extra 0 in the beginning of the string to convert it into multiple of 16.

LSB replacement

The LSB of the first 10 rows of the cover file are replaced by the number of temporary matrix. From row number 11 select each temporary matrix and replace each LSB of the cover file by the location of the nonzero element present in that matrix. After embedding the whole encrypted message into the cover file , the required stego file will be created.

Detailed method for decryption

The stego file is the input to the receiving end.In two phases,the operation is performed in the receiving end. In first phase the required reduced string has been constructed by temporary matrix construction. In second phase the Huffman decryption algorithm has been used to find the required target string.

Digital encoding of the stego file

It performs bit level manipulation to encode the message. The following steps are

- Accept the stego file as input.
- Finds its binary equivalent.
- Block it into 8-bits pattern.

Size estimation

The LSB of the first 10 rows of the stego file contains the number of temporary matrix. In this phase the number of temporary matrix has been constructed from the LSB portion.

Temporary matrix construction

A temporary 4*4 matrix has been constructed where all the elements are initially 0. After row number 10, each 4 consecutive rows first define the number of nonzero elements containingin the temporary matrix and the next part defines the row and column number of each nonzero element. After identifying the row and column number of each nonzero element replace the corresponding position of the temporary matrix by 1. The matrix construction is shown in Figure 3.

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 3. Temporary matrix in the receiving side

Now let us consider the LSB of the four sequential consecutive blocks are 0111, 0001,0010,0110,0111,1001,1010,1101.The decimal equivalent of the first four LSB is 7, which signifies the number of nonzero elements in the temporary matrix . For the rest of the elements, two left most bit signifies the row number and two right most bit signifies the corresponding column number of the nonzero element. The location of the nonzero elements isshown in Table 4.

Table 4. Location identification of the nonzero elements

Two left most bits	Row number	Two right most bits	Column number
00	0	01	1
00	0	10	2
01	1	10	2
01	1	11	3
10	2	01	1
10	2	10	2
11	3	01	1

The row and column number identified from table 4 are the location of the nonzero elements. The matrix construction from the nonzero elements isshown in Figure 4.

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 4. Construction of temporary array with nonzero element

Construction of reduced target string from the temporary matrix

All the content of the temporary matrix are stored in a temporary array successively. After completing the entire temporary matrix, the required target string will be created.

Construction of original target string

From the reduced target string the desired target string is constructed by using Huffman decryption algorithm. This has been performed by two steps

- By using the private key the Huffman dictionary has been constructed in the same way in the sender side.
- By using the Huffman dictionary the required target string has been constructed.

Algorithm Encryption

Input: A cover audio file and a target string.
Output: A Stego file.

1. Start
2. Digitalize the cover file and group it into 8 bits block.
3. Input the target string.
4. Construction of private key which is basically a pangram string consists of all the characters and special symbol.
5. Probability calculation of each characters of the private key using algorithm calculates probability.
6. Construction of Huffman tree using algorithm Huffman_Tree.
7. Using Huffman tree convert all the characters of the target string into code words which is the encrypted and reduced version of the original target string.
8. The target string should be multiple of 16 if not convert it into multiple of 16 by padding some extra zeros at the start of the string.
9. Store 16 characters of the target string into a4*4 temporary matrix in row major order.
10. This temporary matrix construction procedure will continue until the end of the target string.
11. From each 4*4 temporary matrix create a temporary table which consists of each nonzero element's row and column number.
12. Each temporary matrix contains 16 elements.So the number of temporary matrix is (No of characters present in the target string)/16. Suppose the number of temporary matrix is temp.
13. Replace the LSB of row number 1-10 of the cover file with the binary representation of temp.
14. For each temp performs the following steps:
    - Replace the next four consecutive LSB of the cover file by the binary equivalent of nonzero elements contains into the temporary matrix constructed from the target string. Suppose the number is num.
    - For each num performs the following steps:
    - Replace the next four consecutive LSB of the cover file by the row and column number of each nonzero element one by one.
15. The stego file has been created.
16. Stop

Algorithm Decryption

Input: The stego file.
Output: The target string.

1. Start.
2. Accept the stego file.
3. Finds the binary equivalent of stego file and group it into 8 bits block.
4. From the LSB of row number 1-10 find the number of temporary matrix constructed from the target string. Suppose the number is temp.

5. Construct a 4*4 temporary matrix (A) consist of all zeros.
6. For each value temp performs the following steps:
7. The decimal equivalent of the LSB of the next four consecutive rows gives the number of nonzero elements into the temporary matrix. Suppose the number is num.
8. For each value of num performs the following steps:
   - From this point the LSB of each four consecutive row of the cover file gives the row and column number of each nonzero element present in the temporary matrix. The two most significant bits identified row number and the two least significant bits identified the column number.
   - Find the row and column number of each nonzero element.
   - Replace the corresponding row and column of the temporary matrix (A) by 1 and rest are zero.
   - The whole reduced target string is constructed form the temporary matrix sequentially and store in an array temp_reduced.
9. Private Key construction in the receiving side.
10. By using the private key the original target string has been constructed form temp_reduced.
11. Stop.

RESULTS

In result analysis phase we have used two sets of cover file: cover file 1 and cover file 2 shown in figure 5 and figure 9 .Three different length datasets has been used for transmission purpose. After embedding the target string into cover file 1 and cover file 2 the corresponding stego file are shown in Figure 6 to 8 and Figure 10 to 12.

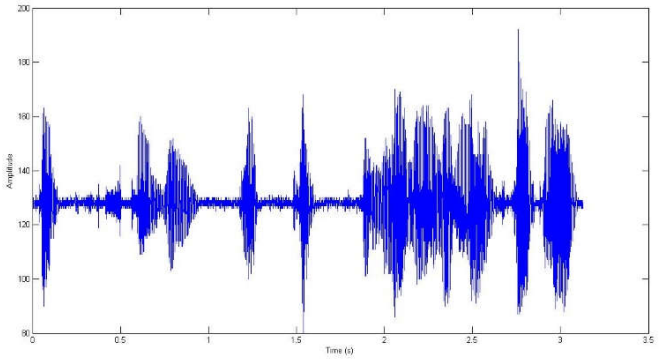

Figure 5. Cover File 1

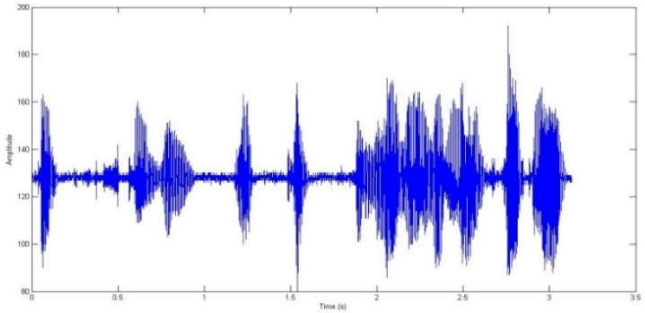

Figure 6. Cover File 1 after embedding 440 char(s)

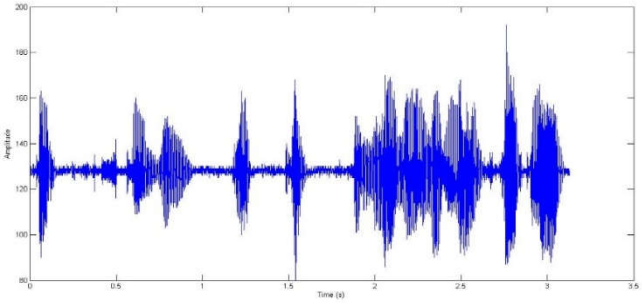

Figure 7. Cover File 1 after embedding 350 char(s)

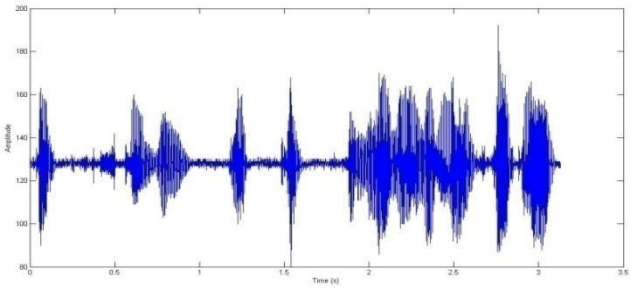

Figure 8. Cover File 1 after embedding 250 char(s)

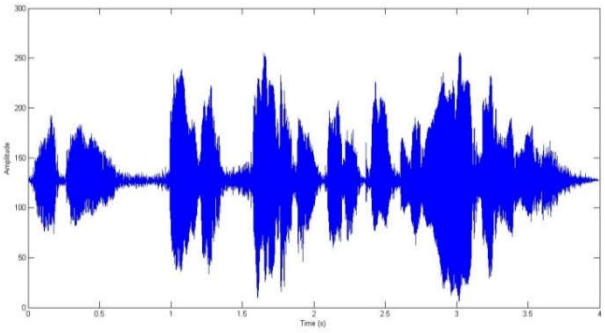

Figure 9. Cover File 2

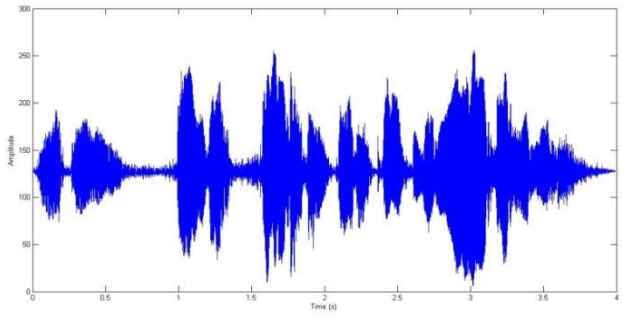

Figure 10. Cover File 2 after embedding 200 character(s)

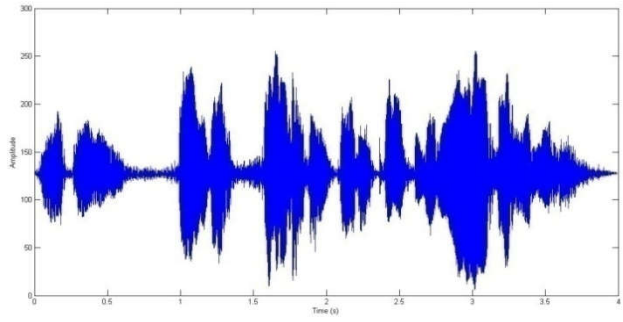

Figure 11. Cover File 2 after embedding 300 character(s)

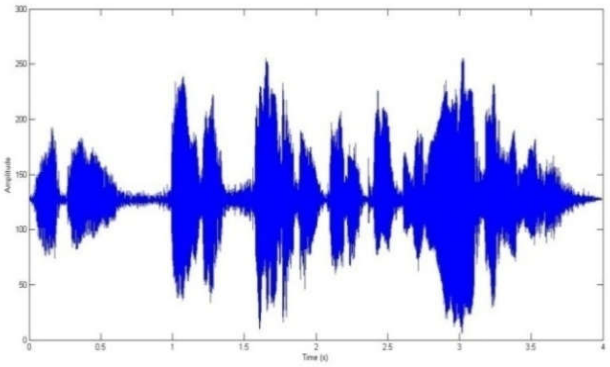

Figure 12. Cover File 2 after embedding 400 character(s)

Imperceptibility and undetectability are the two major criteria of any steganography algorithm. The above graph illustrates that after embedding small, medium and large datasets, the stego file are perceptually indistinguishable from the original cover file. Their sound isalso audibly same. In the next section, we have compared our proposed method with traditional LSB method by twoquality measure Mean square error (MSE) and Signal to Noise Ratio (SNR). The comparison is shown in Table 5.

Table 5.Comparison with standard LSB method

Cover File	No of Character		Huffman & Sparse	Normal LSB
1	440	MSER	0.0271	0.0256
		PSN Ratio	57.8334	58.0869
1	350	MSER	0.0211	0.0206
		PSN Ratio	58.9178	59.0266
1	250	MSER	0.0154	0.0147
		PSN Ratio	60.2964	60.5019
2	300	MSER	0.0288	0.0280
		PSN Ratio	57.6778	57.8098
2	400	MSER	0.0382	0.0373
		PSN Ratio	56.4628	56.5648
2	200	MSER	0.0188	0.0191
		PSN Ratio	59.5356	59.4679

The above table illustrates that the Mid square error (MSER) and Signal to noise ratio (SNR) value of our proposed method is nearly equal to standard LSB method. At the same time, it provides a strong encryption. In normal LSB method,there is no security of data; the data is openly padded into the LSB of the cover file. But in first level , our proposed method encrypt and compress the target string by Huffman coding and in second level only the location of the nonzero elements has been padded.

Conclusion

We have proposed a novel audio steganography method by using the concept of Huffman coding and sparse matrix. A huge volume of text can be embedded into the cover file by using this method. By using a predefined private key, the Huffman encoding has been performed and the secret messages are embedded into non-zero elements of the sparse representation. The experimental results havebeen qualified into two quality measures:Mid square error and Signal to noise ratio. The experimental result shows that difference between the original cover file and the stego file is minute. When compared with traditional LSB method, the value of the Mid square error and Signal to noise ratio shows that it is nearly equal to LSB method. So we can conclude that an improved data security has been obtained by our proposed method.

REFERENCES

Ahani, Soodeh, Shahrokh Ghaemmaghami, and Z. Jane Wang. 2015. "A sparse representation-based wavelet domain speech steganography method."Audio, Speech, and Language Processing, IEEE/ACM Transactions on 23.1 (2015): 80-91.

Banerjee, Sean, Sandip Roy, M. S. Chakraborty, and Simpita Das. 2013. "A variable higher bit approach to audio steganography." *International Conference on In Recent Trends in Information Technology* (ICRTIT), pp. 46-49. IEEE, (2013, Jul 25-27), Chennai, India.

Bhowal, K., Bhattacharyya, D., Pal, A.J., Kim, T.H. 2013. "A GA based audio steganography with enhanced security." Telecommunication Systems., Apr 1; 52(4):2197-2204.

Chao, Min-Wen, *et al*. 2009. "A high capacity 3D steganography algorithm."Visualization and Computer Graphics, IEEE Transactions on 15.2 (2009): 274-284.

Datta, Biswajita, Souptik Tat, and Samir Kumar Bandyopadhyay. 2015. "Robust high capacity audio steganography using modulo operator."International Conference onComputer, Communication, Control and Information Technology (C3IT), IEEE, (2015, December 21-24), Himachal Pradesh, India.

Kiah, M.L., Mat, B. B. Zaidan, A. A. Zaidan, A. Mohammed Ahmed, and Sameer Hasan Al-bakri. 2011. "A review of audio based steganography and digital watermarking." *Int. J. Phys. Sci*., 6, no. 16 (2011): 3837-3850.

Malik, Hafiz, K. P. 2012. Subbalakshmi, and Ramamurti Chandramouli. "Nonparametric steganalysis of qim steganography using approximate entropy." Information Forensics and Security, IEEE Transactions on 7.2 (2012): 418-431.

Rahim, L.B., Bhattacharjee, S. and Aziz, I.B. 2014. "An Audio Steganography Technique to Maximize Data Hiding Capacity along with Least Modification of Host". In Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) Jan 1 (pp. 277-289). Springer Singapore.

Roshidi Din, Hanizan Shaker Hussain, and Sallehuddin Shuib, —"Hiding secret messages in images: suitability of different image file types", Wseas Tionsransac on Computers, vol. 6(1), January 1 2006, pp. 127 -132.

Satish, K., *et al*. 2004. "Chaos based spread spectrum image steganography."Consumer Electronics, IEEE Transactions on 50.2 (2004): 587-590.

Vimal, Jithu. "Literature Review on Audio Steganographic Techniques."

Zamani, M., Manaf, A., Ahmad, R.B., Jaryani, F., Taherdoost, H, Zeki, A.M. 2009. "A secure audio steganography approach". International Conference on Internet Technology and Secured Transactions, (ICITST) (2009 Nov 9) pp. 1-6. IEEE, London, UK.

****