International Journal of Security and Its Applications
Vol. 10, No. 2 (2016), pp.51-62
http://dx.doi.org/10.14257/ijsia.2016.10.2.05

# A View on LSB Based Audio Steganography

Ratul Chowdhury [1], Debnath Bhattacharyya[2],
Samir Kumar Bandyopadhyay[3] and Tai-hoon Kim[4]

*[1]Department of Information Technology,*
*Institute of Engineering and Management, Kolkata, India*
*[2]Department of Information Technology,*
*Bharati Vidyapeeth University College of Engineering, Pune-411043, India*
*[3]Department of Computer Science and Engineering,*
*University of Calcutta, Kolkata, India*
*[4]Department of Convergence Security, Sungshin Women's University,*
*249-1, Dongseon-dong 3-ga, Seoul, 136-742, Korea*
*[1]ratul.chowdhury@iemcal.com, [2]debnathb@gmail.com,*
*[3]Skb1@vsnl.com, [4]taihoonn@daum.net*

## Abstract

*In this paper the concept of cryptography and steganography are combined to perform a powerful encryption. Here we propose a novel approach where a duel encryption methodology has been implemented. In the first level of encryption a pattern matching algorithm has been employed to encrypt the text message in terms of their positional value. In second level, the conventional LSB method has been used to embed the positional value in the cover file. Such a duel encryption method will ensure data security in an efficient manner. Finally the performance of the proposed method is evaluated in terms of means square error (MSE) and signal to noise ratio (SNR). A comparison has been carried out with conventional LSB method. The experimental results and the comparisons demonstrated that our algorithm is highly efficient in terms of encryption and the capacity size of the text.*

***Keywords****: Audio Steganography, ASCII, blocking, Pattern matching, least significant bit technique (LSB)*

## 1. Introduction

Information security, sometimes shortened to InfoSec, is the technique of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction.

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 word Steganography was derived from ancient Greek words steganos meaning "covered, concealed, or protected" and graphein meaning "writing”. It is a technique to hiding of a message within another so that the presence of the hidden message is indiscernible.

Unlike cryptography the message is unaltered in steganography. Here the messages are hidden within a media in such a way so that none can understand the very existence of the message *i.e.* it cannot be perceived by human. The combination of cryptography and steganography provide two level of security.

ISSN: 1738-9976 IJSIA

The first use of steganography is reported to be used by Herodotus, who wrote hidden message on a tablet and covered it with wax. The embedding of secret letter in the messengers' shoe soles or women's ear rings is the another application in ancient world.

Least Significant Bit (LSB) coding is the simplest way to embed information in a digital audio file by replacing the least significant bit of each sampling point with a binary message [2].Main advantages of LSB coding is that it allows a large volume of data given in audio or text format to be encoded and data are found in the receiving end in loss-less format [5 and 7].

In this paper we combine cryptography and steganography to perform a powerful encryption. In this proposed method, a cover file and a text file are used as an input .By using a pattern matching algorithm the encrypted version of the text is embedded into the cover file and a stego file is generated. In the receiving end by applying a reverse pattern matching algorithm the original text is fetched form the encrypted file.

## 2. Review Works

Someone takes the first letter of each word of the previous sentence to see that it is possible and not very difficult. Hiding information in plain text can be done in many different ways [8, 9, and 13]. Many techniques involve the modification of the layout of a text, rules like using every n$^{th}$ character or the altering of the amount of white space after lines or between words. The last technique was successfully used in practice and even after a text has been printed and copied on paper for ten times, the secret message could still be retrieved. Another possible way of storing a secret inside a text is using a publicly available cover source, a book or a newspaper, and using a code which consists for example of a combination of a page number, a line number and a character number. This way, no information stored inside the cover source will lead to the hidden message. Discovering it relies solely on gaining knowledge of the secret key.

There are some advanced LSB method where encrypted version of the text are embedded in to the cover file for better encryption. In 2015 Prof Samir Kumar Bandyopadhyay and Biswajita Datta used the modulo operator to perform a duel encryption method there the first level encryption is done by using modulo operator and in second level by using standard LSB method they embed the encrypted version of the text into the cover file [12].The algorithm from different domains are currently being used to perform better encryption [3]. Genetic algorithm base approach is another example of duel encryption. [1, 6].

An alternative approach is to perform the first level encryption by using XOR operation and then message embedding by using standard LSB method [10 and 14].Another method is spectrum manipulation where the frequency of the transmitted signal is deliberately varied to perform better encryption [10].

There are lots of methods related to image encryption where a message is secretly hidden within an image [5]. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in noisy areas that draw less attention—those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. A number of ways exist to hide information in digital media. Common approaches include

- Least significant bit insertion.
- Masking and filtering.
- Redundant Pattern Encoding.
- Encrypt and Scatter.
- Algorithms and transformations.

Least significant bit (LSB) insertion is a common and simple approach to embed information in a cover file. In this method the LSB of a byte is replaced with an M’s bit. This technique works better for image, audio and video steganography. To the human eye, the resulting image will look identical to the cover object.

DCT is a lossy compression transform because the cosine values cannot be calculated exactly, and repeated calculations using limited precision numbers introduce rounding errors into the final result. Variances between original data values and restored data values depend on the method used to calculate DCT.

Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media, such as digital images. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced. These methods range from rather simple algorithms that insert information in the form of signal noise to more powerful methods that exploit sophisticated signal processing techniques to hide information. The list of methods that are commonly used for audio steganography is given below [13, 16 and 17]

- LSB coding
- Parity coding
- Phase coding
- Spread spectrum
- Echo hiding

**Parity Coding:** The parity coding method breaks a signal down into separate regions and encodes each bit from the secret message in the sample region's parity bit.

**Phase Coding:** Since phase components of sound are less perceptible to the human ear than noise so the phase of an initial audio segment is substituted with a reference phase that represents the data.

**Echo data hiding:** Secret information can be embedded in audio data by introducing an echo to the original signal and then the data is hidden by varying three parameters of the echo: initial amplitude, decay rate, and offset.

**Spread Spectrum (SS):** The SS method spreads the secret message over the frequency spectrum of sound file which is independent of the actual signal.

## 3. Detailed Method for Encryption

### 3.1. Digital Encoding of the Cover File and the Message File

The cover audio file in the form of bytes is first converted it into 8 bit patterns. After digital encoding the cover file has n rows and 8 column. Next Receives the message file into text format. Convert each character of the text file into 8 bits binary according to the ASCII value of the character.

### 3.2. Blocking and Pattern Matching

a. The message file is divided character wise.
b. Conversion of each alphabet to its equivalent ASCII codes.
c. The ASCII codes are converted into its binary form.

As per example suppose there are 3 a, b, c characters in the text message. There digital encoding and 8 bits blocking are given in Figure. 1

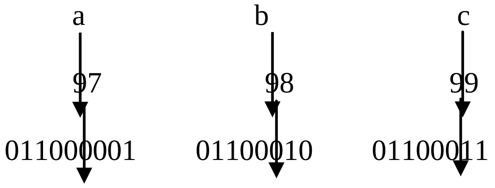


**Figure 1. Digital Encoding and Blocking**

d. After convert the message into binary format, the first two blocks of the cover file has to select for pattern matching suppose the first two 8 bits block of the cover file are copied into an array index from 1 to 16 shown in Figure 2.

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

**Figure 2. Pattern Matching Array**

e. Then each 8 bits ASCII of each alphabet of the message file is divided into 2 bits block.

From the example 8 bits ASCII of the message audio file are given below:
01100001
01100010
01100011

Now, Let us consider the first 8 bit ASCII of the first alphabet of the message file. It has four two bits block. (01, 10, 00, 01)

f. This phase will match the pattern of each two bits block of the message file from the pattern matching array and return the first matching location. Shown in Figure 3.

| **1** | **2** | 3 | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

**Figure 3. Pattern Matching and Location Identification**

From the above diagram we get,

**00** → pattern is in location **7** → 0111
01 → pattern is in location **1** → 0001
10 → pattern is in location **3** → 0011

Each matching decimal location of the array is then converted into 4 bit binary form. For each 8 bits block it is the same.

### 3.3. Size of the Encrypted Text File

This phase will estimate the size of the text message file

| | | | |
|---|---|---|---|
| 01 | 10 | 00 | 01 |
| 1 | 3 | 7 | 1 |
| 0001 | 0011 | 0111 | 0001 |

**Figure 4. Size Estimation**

Here LSB substitution is used, so it will add each 0001, 0011, 0111, and 0001 into the LSB of each row of the cover file described in Figure 4. Total 16 rows are required to represent one character of the text message. If there are ‘n’ number of characters in the text message then total 16*n number of rows of the cover file are required to accommodate the whole text. Here, ‘n’ decimal values are represented into 20 bits binary.

### 3.4. Pattern Matching Probability

For a 2-bit sequence, the number of possible combinations will be $2^2$ =4.So for a properly randomized sample, that is if we choose a 4-bit sequence and try to locate a given 2-bit pattern within it then probability of finding the 2 bit pattern at any location is ¼*4=1.Here we have used 16 bits pattern for sample space so it is 4 times larger than the minimum necessary size to fulfil the above criteria. So we will always be able to find the 2-bit sequence within the 16 bit pattern.

### 3.5. LSB Replacement

a. The first two rows of the cover file are used to pattern matching.
b. In LSB of the next 20 rows the cover file the size of the text message is embedded.
c. In LSB of the next 16*n rows are used to accommodate encrypted version of the text message.
d. After embedding the whole encrypted message into the cover file the required stego file will create.

## 4. Detailed Method for Decryption

In receiving side the required stego file is the input. And the receiver apply the reverse procedure to decrypt the text from them.

### 4.1 Digital Encoding of the Stego File

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

a. Accept the stego file as input.
b. Digitalize it.
c. Block it into 8-bits pattern.

### 4.2. Rows Estimation

Store the first two 8 bits block of the cover file into an array for pattern matching .From the next 20 rows select the LSB for size estimation. Convert it into decimal form and after multiplying the decimal value with 16 the exact row number will produce where the text in hidden.

### 4.3. Matching Pattern and Hidden Message Identification

Let us consider the 16 corresponding LSB bits of the cover file are 0001001101110001. The four bits grouping, its corresponding decimal value and the steps to identify the hidden message are shown in Figure 5.The pattern matching array in the receiving side is shown in Figure 6.

| | | | | |
|---|---|---|---|---|
| Four bit grouping | 0001 | 0011 | 0111 | 0001 |
| Decimal | 1 | 3 | 7 | 1 |
| Two consecutive bits | 01 | 10 | 00 | 01 |
| Hidden message | 01100001 | | | |

**Figure 5. Blocking and Hidden Message Identification**

This process will repeat until the whole text message is fetched from the cover file.

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

**Figure 6. Pattern Matching Array in Receiving Side**

## 5. Detailed Algorithm

### 5.1. Algorithm Encryption

Input: A cover file and a text file
Output: A stego file where the encrypted version of the text is hidden.
1. Start
2. Read the text file and cover audio file.
3. Convert the cover audio file into digital form which contains n rows and 8 columns.
4. Convert the text message into digital form where each character is represented by 8 bits according to the ACSII value of that character present in the text.
5. Find the length of the text message.
6. Store the first two column of the cover file into an array index from 1 to 16 for pattern matching.
7. Group the digital representation of the text message into 2 bits block
8. Represent the size of the text message into 20 bits binary.
9. From row number 3 to 22 replace the LSB of the cover file by the length of the text message represented in binary form.
10. Search each two bits pattern of the text message from the pattern matching array and find its matching location .If there are multiple match select the first matching location.
11. Convert the matched location into 4 bits binary.
12. From row number 23 replace all the LSB of the cover file by these matched location.
14. This process will continue until the last character of the text message.
15. END

### 5.2. Algorithm Decryption

Input: The stego file.
Output: The original text file.
1. Start
2. Read the stego file into the receiving end.
3. Find the digital equivalent of the stego file.
4. Group it into 8 bits block and it has n rows and 8 column.
5. Store the first two rows of the stego file into an array index from 1 to 16 for pattern matching.

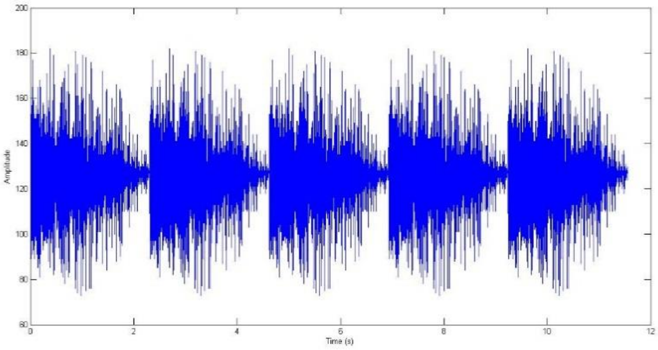
6. From row number 3 to 22 store all the LSB of the cover file and find its decimal equivalent.
7. Multiply 16 with these decimal value to find the row the number where the text message in hidden into the cover file.
8. From row number 23 to the next 16 rows select all the LSB of the cover file step by step.
9. Group it into 4 bits block.
10. Each decimal value of the 4 bit block identifies the array index which gives the starting address of a 2 bit blocks of the text message.
11. According to these array index receiver will fetch 2 consecutive bits from the pattern matching array and four two bits block will create the binary equivalent of one character of the text message.
12 .Find the decimal equivalent of each 8 bits block which identifies the ASCII value of one single character of the text message.
13. Repeat steps 8 to 12 until the end of the text file hidden in cover file.
14. End

## 6. Results

In this section first the waveform of the experimental results are shown from Figure 7 to Figure 14. In Figure 7 and Figure 11 two different cover files cover1.wav and cover2.wav has been used. Three sets of target text is chosen (small, medium, large) for embedding into the cover file.

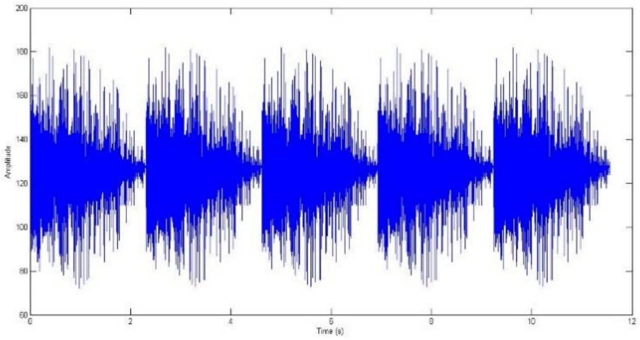
According to the graph it is observed that original cover file and the corresponding encrypted file is reasonably same and their sound also audibly same .There is no way to identify the existence of the message into the cover file. In the decryption end by applying reverse pattern matching algorithm the message is decrypted from the stego file which is exactly equal with the input text and its format also equal. So it is concluded that it is lossless pattern matching algorithm.

Now the qualities of the experimental results are analyzed by two parameters Means square error calculation and SNR calculation for both cover and stego audio with existing relevant methods like standard LSB technique shown in Table 1.



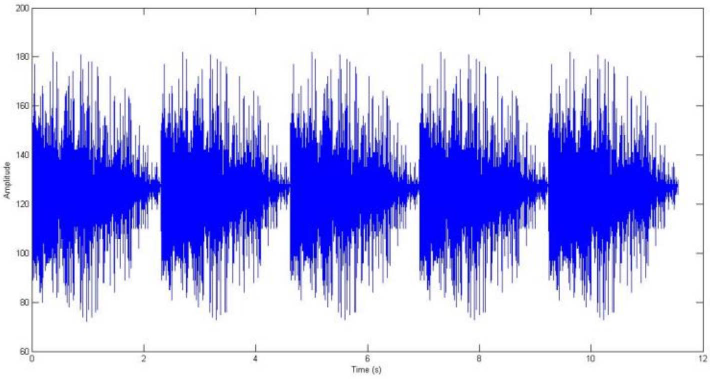
Cover File:- Cover1.wav

**Figure 7. Original Cover File:- *cover1.wav***



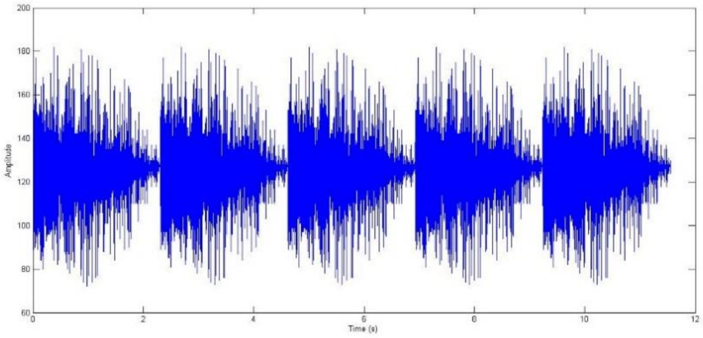
Small Dataset:-Message Length *48*

**Figure 8. Encrypted File:-** ***encrypted1.wav***



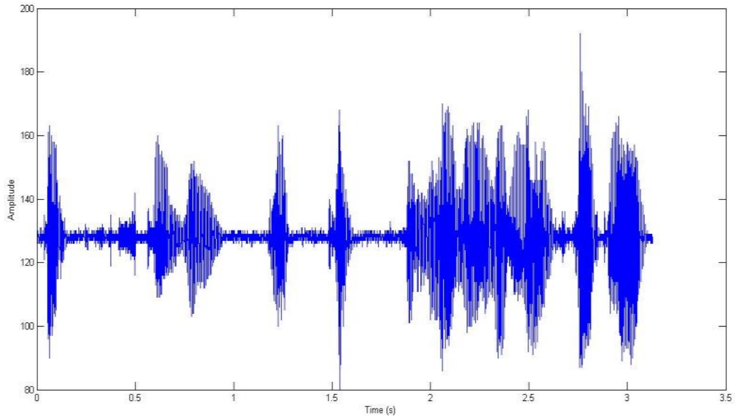
Medium Dataset: - Message Length: - *160*

**Figure 9. Encrypted File:-** ***encrypted2.wav***



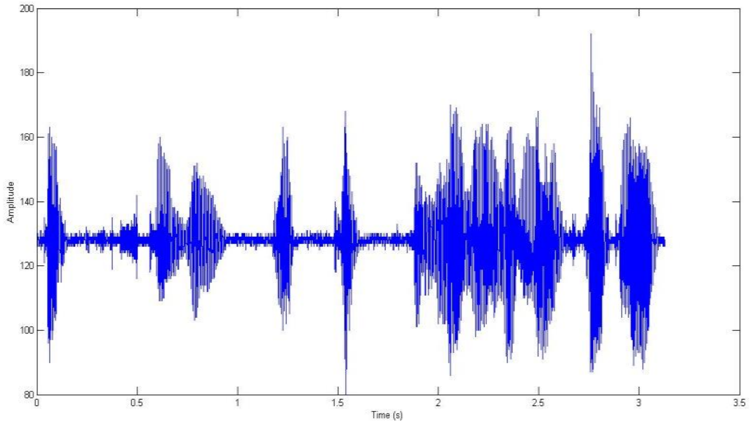
Large Dataset: - Message Length: - *240*

**Figure 10. Encrypted File:-** ***encrypted3.wav***



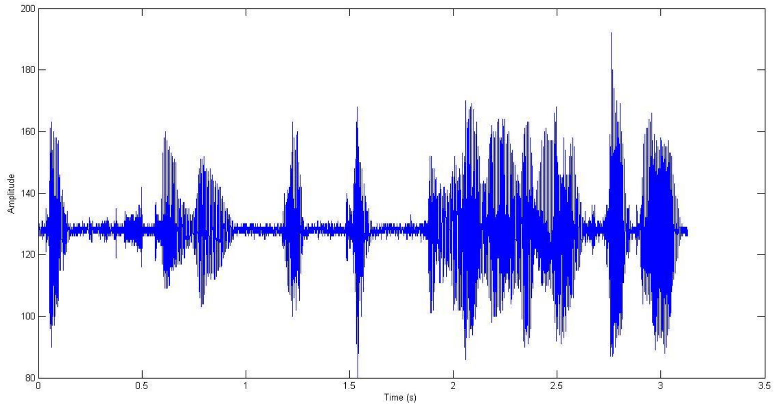
Cover file:- Cover2.wav

**Figure 11. Original Cover File:-** ***cover2.wav***



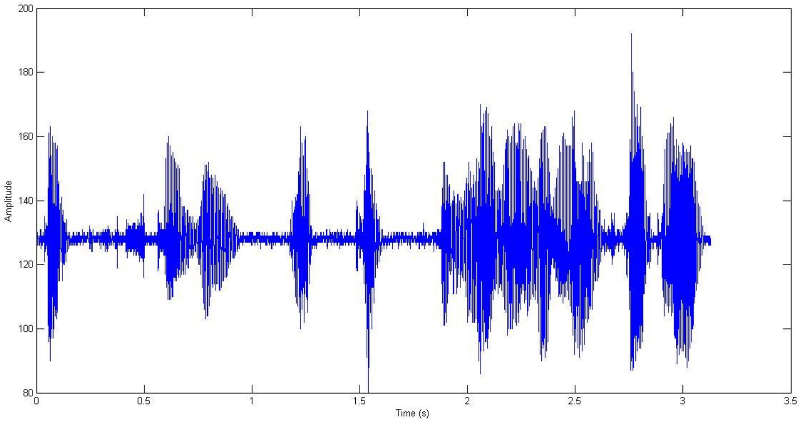
Small Dataset: - Message Length 7*4*

**Figure 12. Encrypted File:-** ***encrypted1.wav***



Medium Dataset: - Message Length 228

**Figure 13. Encrypted File:-** ***encrypted2.wav***



Large Dataset: - Message Length 302

**Figure 14. Encrypted File:- *encrypted3.wav***

**Table 1. Comparison with standard LSB method**

| Cover File | Length of Message | Normal LSB Method | | Pattern Matching Method | | |
|---|---|---|---|---|---|---|
| | | MSER | SNR | MSER | SNR | |
| 1 | 50 | 0.0019 | 69.0869 | 0.0047 | 65.2248 | |
| 1 | 100 | 0.0040 | 65.9811 | 0.0094 | 65.2544 | |
| 1 | 200 | 0.0083 | 62.7916 | 0.0188 | 59.2416 | |
| 2 | 70 | 0.1102 | 51.8635 | 0.1546 | 50.3955 | |
| 2 | 140 | 0.2316 | 48.6383 | 0.3023 | 47.4817 | |
| 2 | 250 | 0.3994 | 46.2726 | 0.5619 | 45.1493 | |

As compared with LSB method we say that it is a well-known technique and there is no security of data in this technique. The whole text message is embedded into the LSB of the cover file without any encryption. So the intruder can easily encrypt the message from the cover file. But our pattern matching method uses a duel encryption strategy and for encryption purpose we have increased number of bits of the text message as a result the Means Square has increased and SNR value has decreased from LSB method. Although there is an increased in means square error  compared with LSB method however an improved data security has been obtained in this method in other word the cost of enhanced data security has been obtained as an increasing Mean square error and decreasing SNR.

## 7. Conclusion

In this paper a high quality duel encryption methodology has been implement. By using this method the encrypted version of a large volume of text can be embedded into the cover file and it is accepted in the receiving end without any change. So it is concluded that the integrity and quality of the message are well maintained. The provided result have conformed this conclusion.

The developed algorithm can be extended to have lesser bandwidth requirement by reducing the number of bits of the cover file. Different data compression algorithm can also be exposed with our work to accommodate a large version of text into a cover file.
Finally it may be stated that the algorithm discussed in this paper has duel encryption capability however the encryption achieved is lossless and integrity and quality of the data also well maintained.

## References

[1] P. Johri, and A. Kumar, "Review paper on text and audio steganography using GA", International Conference on Computing, Communication & Automation (ICCCA), Uttar Pradesh, India, (**2015)**, May 15-16, pp. 190-192.

[2] V. Jithu, and A. Mary Alex. "Audio steganography using dual randomness LSB method." International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), IEEE, **(2014)**, Jul 10-11, Tamilnadu, India, pp. 941-944.

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

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

[5] R. Din, H. Shaker Hussain, and S. Shuib, ―"Hiding secret messages in images: suitability of different image file types", WSEAS TIONSRANSAC *on* COMPUTERS, vol. 6, no. 1, January 1 **(2006)**, pp. 127 -132.

[6] K. Bhowal, D. Bhattacharyya, AJ Pal, TH Kim, "A GA based audio steganography with enhanced security." Telecommunication Systems. **(2013)** Apr 1, vol. 52, no. 4, pp. 2197-2204.

[7] LB Rahim, S. Bhattacharje and IB Aziz, "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) **(2014)** Jan 1, pp. 277-289, Springer Singapore.

[8] Balgurgi, P. Pooja, and S. K. Jagtap. "Audio steganography used for secure data transmission." In Proceedings of International Conference on Advances in Computing. Springer India, **(2012)**, pp. 699-706

[9] R. J. Anderson (ed.), "Information hiding", 1st international workshop, volume 1174 of Lecture Notes in Computer Science, Isaac Newton Institute, Springer-Verlag, Berlin, Germany, **(1996).**

[10] Nathan, Mark, N. Parab and K. T. Talele. "Audio Steganography Using Spectrum Manipulation." In Technology Systems and Management, Springer Berlin Heidelberg, **(2011**), pp. 152-159.

[11] S. Malviya, M. Saxena, A. Khare, "Audio Steganography by Different Methods", International Journal of Emerging Technology and Advanced Engineering [20] (ISSN 2250-2459, vol. 2, issue 7. **(2012)**.

[12] Datta, Biswajita, S. Tat, and S.Kumar Bandyopadhyay. "Robust high capacity audio steganography using modulo operator." International Conference on Computer, Communication, Control and Information Technology (C3IT), IEEE, **(2015)** December 21-24, Himachal Pradesh, India.

[13] S.K Bandyopadhyay, D. Bhattacharyya, D. Ganguly, S. Mukherjee, P. Das, "A tutorial review on steganography". In International conference on contemporary computing, **(2008)** Aug 7, vol. 101.

[14] R. Radhakrishnan, M Kharrazi and N. Memon, "Data masking: A new approach for steganography?" Journal of VLSI signal processing systems for signal, image and video technology. **(2005)** Nov 1, vol. 41, o. 3, pp. 293-303.

[15] R. Darsana, A. Vijayan, "Audio steganography using modified LSB and PVD". In Trends in Network and Communications , Springer Berlin Heidelberg, **(2011)** Jan 1, pp. 11-20.

[16] R. J. Anderson (ed.), "Information hiding", 1st international workshop, volume1174 of Lecture Notes in Computer Science, Isaac Newton Institute, Springer-Verlag, Berlin, Germany,**(1996)**.

[17] Petit colas FA, Anderson RJ, Kuhn MG. "Information hiding-a survey." Proceedings of the IEEE. **(1999)**, vol. 87, no. 7, pp. 1062-78.