Complex Encryption Methodology Based On Two Level Encryption Technique

Samir Kumar Bandyopadhyay
University of Calcutta
India
skb1@vsnl.com

ABSTRACT: *In this paper a complex encryption methodology has been employed to perform a high level data security. Here a two level encryption technique has been described. In first level a pattern matching array has been created in such a way such that all possible combination of any 4 bits sequences are present there. By using this pattern matching array the positional value of the secret message has been identified and in second level the positional value has been embedded into the LSB of the cover file. This technique is independent of the size of the text message and message retrieved in the receiving end is totally lossless .The quality of the proposed method is evaluated by using two factors Means square error(MSE) and signal to noise ratio(SNR) and finally a comparison has been done with different traditional steganography techniques. According to different experimental results and the comparison our method is extremely efficient in terms of encryption and the size of the text.*

Keywords: HVS, HAS, SS and DWT

Received: 14 April 2017, Revised 1 June 2017, Accepted 8 June 2017

1. Introduction

Steganography is the art and science of embedding hidden information in such a way that no one, apart from the sender and intended recipient, identifies the existence of the message into the cover file. Steganography works by replacing bits of useless or unused data in regular computer Files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different invisible information. This hidden information can be plain text, cipher text, audio or even images.

In a computer-based audio Steganography system, secret messages are embedded in digital audio. The secret message is embedded by slightly altering the binary sequence of the sound file. Existing audio Steganography software can embed messages in WAV, AU, and even MP3 audio files. Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media, such as digital images because human air is very perceptible to noise.[3,4] 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 message.

Cryptography on the other hand scrambles a message into a code to obscure its meaning, these two secret communication technologies are used separately or together—for example, by first level encrypting a message, then hiding it in another file for transmission. As the world becomes more anxious about the use of secret communication, and as regulations are created by

governments to limit uses of encryption so the importance of steganography becomes prominence day by day.

In this paper we combine the concept of cryptography and steganography. In first level the message is encrypted by using a pattern matching algorithm and in second level the message is secretly hidden into a cover file. To perform the pattern matching algorithm a 28 bit sequence array has been employed in such a way that all possible combination of a four bit sequence are present there. The target string is also converted into binary sequence. In next level each four sequence of the target string is searched from the pattern matching array and instead of sending the target string directly we have send its location. This methodology increases the level of encryption in a well manner. For sending purpose the conventional LSB method has been used. Main advantages of LSB coding is that it allows a huge volume of data given in audio or text format to be encoded and data are found in the receiving end in loss-less way.

The message is decrypted in the receiving side in a loss less way. Finally the quality of the proposed method is measured with respect to two parameters mid square error and signals to noise ratio which concluded that there is a very minor change between the original cover file and the stego file so it is very difficult for the intruder to identify the existence of the target string into the stego file.

2. Literature Survey

In audio steganography, the weakness of the Human Auditory System (HAS) is basically used to hide information in the audio file. Because the human auditory system has more precision than Human Visual System (HVS), that's why audio steganography is more challenging than image steganography.

The lists of methods that are commonly used to perform audio steganography are LSB coding. Least significant bit coding is the simplest way to embed information in a digital audio file. 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. [2] The others traditional steganography technique is described below. [7, 8, 11 and 13]

Parity Coding
In these technique an extra parity bits has been employed with the message so that if there is any change in the message it can be easily identified by the receiver.

Phase Coding
Rather than introducing perturbations, the technique encodes the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to- noise ratio.

Echo Hiding
Echo hiding embeds its data by creating an echo to the source audio. Three parameters of this Artificial echo are used to hide the embedded data, the delay, the decay rate and the initial amplitude. As the delay between the original source audio and the echo decrease it becomes harder for the human ear to distinguish between the two signals until eventually a created carrier sound's echo is just heard as extra resonance.

Spread Spectrum (SS)
The Spread spectrum method spreads the secret message over the frequency spectrum of sound file which is independent of the actual signal.[10]

Tone Insertion
This technique is based on insertion of lower power tones in the presence of significantly higher ones. The masking effect is a property of human auditory system (HAS) which make any weak speech component imperceptible by listeners in presence of a much louder one. By inserting tones at known frequencies and at low power level, concealed embedding and correct data of the unintentional attacks such as low pass filtering and bit truncation.

Wavelet Coefficient
In this case, the audio steganography is based on discrete wavelet transform (DWT). Data embedding is done in the LSBs of the

wavelet coefficients achieving high capacity of 200 kbps in 44.1 kHz audio signal. To improve the embedded data imperceptibility, a hearing threshold is introduced and the data hiding in silent parts of the audio signal is avoided. Data hiding in wavelet domain obtains high embedding rate but data extraction at the receiver side might have some errors.

Day by day to increment the level of encryption some different methodology are incorporate with the above conventional method. Instead of sending the original text message the encrypted version of the text are now embedded with the cover audio file. Modulo operator, XOR operation, gray code converter are the technique generally used to perform the first level encryption. [12] Different traditional cryptographic algorithm is directly used for encryption such as RSA algorithm, AES, DES, MD5 algorithm. Algorithm from different domains are now a days used in steganography. Genetic algorithm based approach is a technique where a new sample message has been constructed by using alteration, modification and verification and in the receiving end the reconstruction of the message has been done.[1, 6]

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.

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.

3. Detailed Method of Encryption Process

3.1 Preprocessing

For preprocessing each character of target string converted to their corresponding ASCII as well as 8 bits binary and the cover file is converted into n*8 matrix where n is the number of rows and 8 is the number of columns.

3.2 Creation of Pattern Matching Array

For pattern matching purpose a 28 bits sequence static pattern matching array has been employed, where all possible combination of a four bit sequence are present. Shown in figure 1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	0	0	0	0	1	0	0	1	0	0
15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	0	1	0	0	0	1	0	1	1	1	0	0

Figure 1. Pattern matching array

3.3 Procedure of Pattern Matching

1) The target string is converted in 8 bits binary according to their corresponding ASCII value. As per example suppose the required string is 'abc'. Their bit level manipulation is given in figure 2.

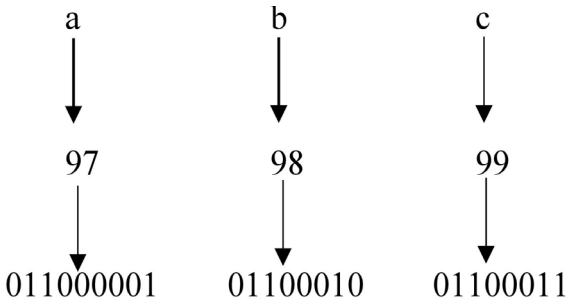


Figure 2. Bit level manipulation of the target string

2) Each 8 bits representation of the target string is divided into two 4 bits block.

From **figure 1** the 8 bits ASCII of the target string are given below:

01100001
01100010
01100011

The first alphabet has two four bits block.

0110,0001

3) The positional value of each 4 bit sequence is identified from the pattern matching array.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	0	0	0	0	1	0	0	1	0	0
15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	0	1	0	0	0	1	0	1	1	1	0	0

Figure 3. Location identification from pattern matching array

4) The positional value indentified in figure 3 is converted into 5 bits binary.

0110 → pattern starts from location 14 → 01110

0001 → pattern starts from location 6 → 00110

3.4 LSB Substitution

Instead of sending the direct target string the encrypted version of it are used for LSB substitution. The positional value of each character identified in figure 3 is embedded into the LSB position of the cover file.

Each 4 bits sequence of the target string is represented into 5 bits positional value, so to represent a 8 bits single character by using this pattern matching algorithm required 10 bits. Each character is employed into the LSB position of each row of the cover file. If the target string has n number of characters then total 10*n number of rows of the cover file are required. For decryption purpose the 28 bits sequence pattern matching array and the length of the target string has to be send from sender side. We

allotted 20 bits to for the length of the target string so maximum 2^{20} characters can be accommodate. The exact LSB substitution process is performed by the following steps.

- The 28 bit sequence pattern matching array is placed into the first 28 rows of the cover file.
- Into the next 20 rows the size of the target string is embedded.
- The next 10*n number of rows of the cover file are used to accommodate the encrypted version of the target string.
- The required stego file is created.

4. Detailed Method for Decryption

At receiving end the stego file is the input and the receiver will perform the following steps to decrypt the message from them.

4.1 Preprocessing of the Stego File

For preprocessing purpose the stego file is represented into an n*8 matrix where n is the number of rows and 8 is the number of columns.

4.2 Identification of the Pattern Matching Array

The LSB values of the consecutive 28 bits of the cover file are stored in an array index from 1 to 28. This array is used further for pattern matching purpose.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	0	0	0	0	1	0	0	1	0	0
15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	0	1	0	0	0	1	0	1	1	1	0	0

Figure 4. Pattern matching array in receiving side

4.3 Rows Estimation

The number of rows of the cover file where the target string is hidden is identified from the next 20 bits. From the next 20 rows select the LSB. Convert it into decimal form and after multiplying the decimal value with 10 the exact row number will produce where the text in hidden.

4.4 Identification of Target String

The target string is identified from the pattern matching array by using the following steps

- Store the consecutive LSB bits of the cover file from row number 49 to 10*n into a temporary array.
- Group it into 5 bits block.
- Find the decimal equivalent of each 5 bits block.
- Each decimal equivalent identifies the starting location of a four bits target string. Starting from this desired location four

	Suppose two 5 bits blocking are	01110, 00110
	Five bit grouping	01110 00110
	Decimal equivalent	14 6
Four consecutive bits from pattern matching array		0110 0001
	Hidden message	01100001 = 'a'

Figure 5. Hidden message identification

consecutive bits of the pattern matching array identifies a part of one character of the target string.

• Two four bits stream creates one character of the target string.

• This procedure will continue until the calculated rows of the cover file and the desired target string is identified into the receiving end. The detailed method is shown in figure 5.

5. Detailed Algorithm

The detailed algorithm for encryption and decryption process is given below.

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 following pattern in the array and replace the LSB of the cover file from row 1 to 28 with the following pattern for pattern matching.

[1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0]

7. Group the digital representation of the text message into 4 bits block

8. Represent the size of the text message into 20 bits binary.

9. From row number 29 to 49 replace the LSB of the cover file by the length of the text message represented in binary form.

10. Search each four 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 5 bits binary.

12. From row number 50 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 LSB of the first 28 rows of the stego file into an array index from 1 to 28 for pattern matching.

6. From row number 29 to 49 store all the LSB of the cover file and find its decimal equivalent.

7. Multiply 10 with these decimal value to find the row the number where the text message in hidden into the cover file.

8. From row number 50 to the next 10 rows select all the LSB of the cover file step by step.

9. Group it into 5 bits block.

10. Each decimal value of the 5 bit block identifies the array index which gives the starting address of a 4 bit blocks of the text message.

11. According to these array index receiver will fetch 4 consecutive bits from the pattern matching array and five 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. Result Analysis

In result analysis phase from figure 6 to figure 13 the wave form of the experimental result are shown. In the next section a comparison has been done with tradition LSB method in respect with two parameters mid square error and signal to noise ratio.

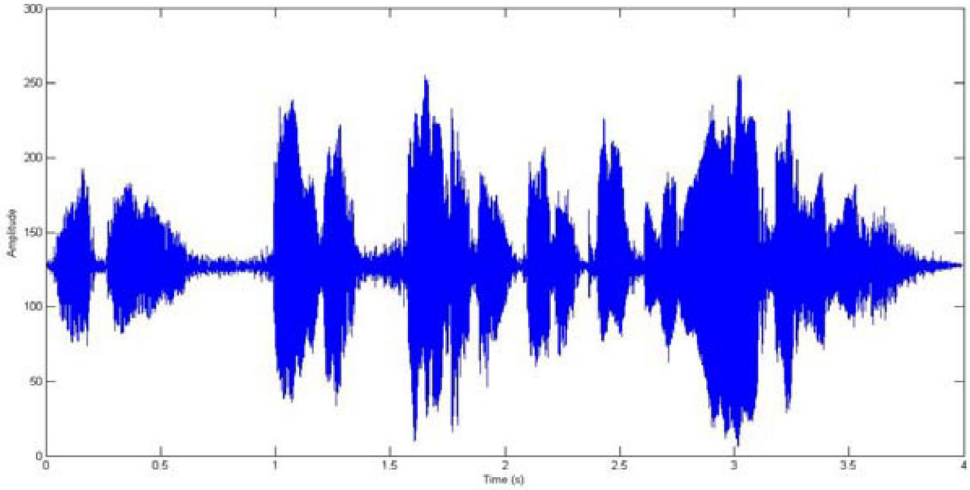
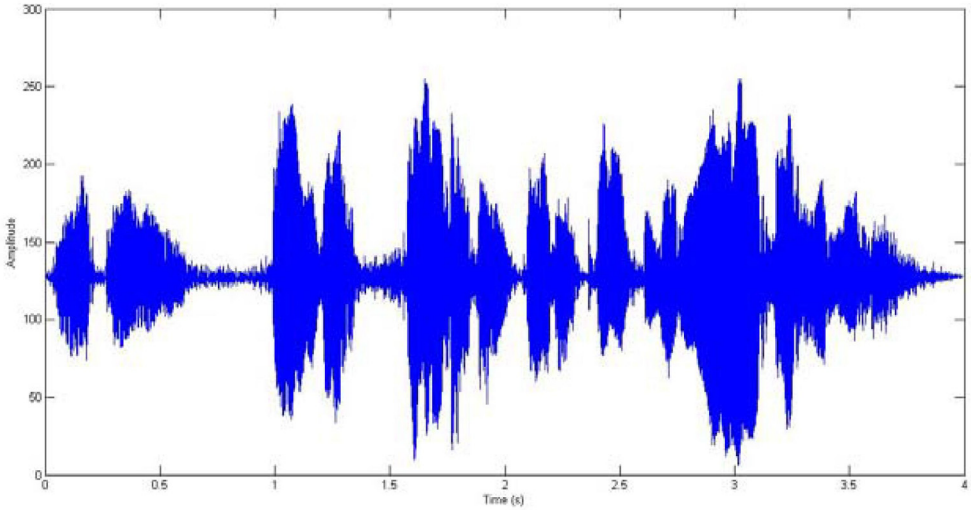


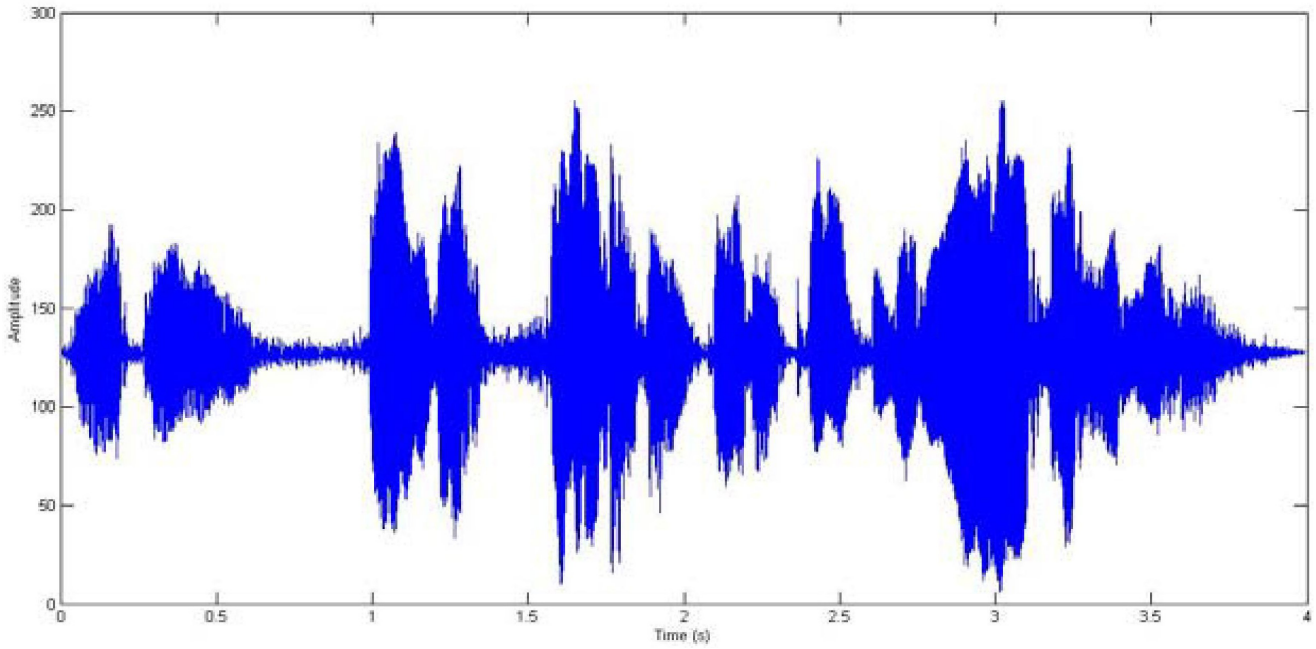
Figure 6. Cover file: *Cover1.wav*



Small Dataset: Message Length 102

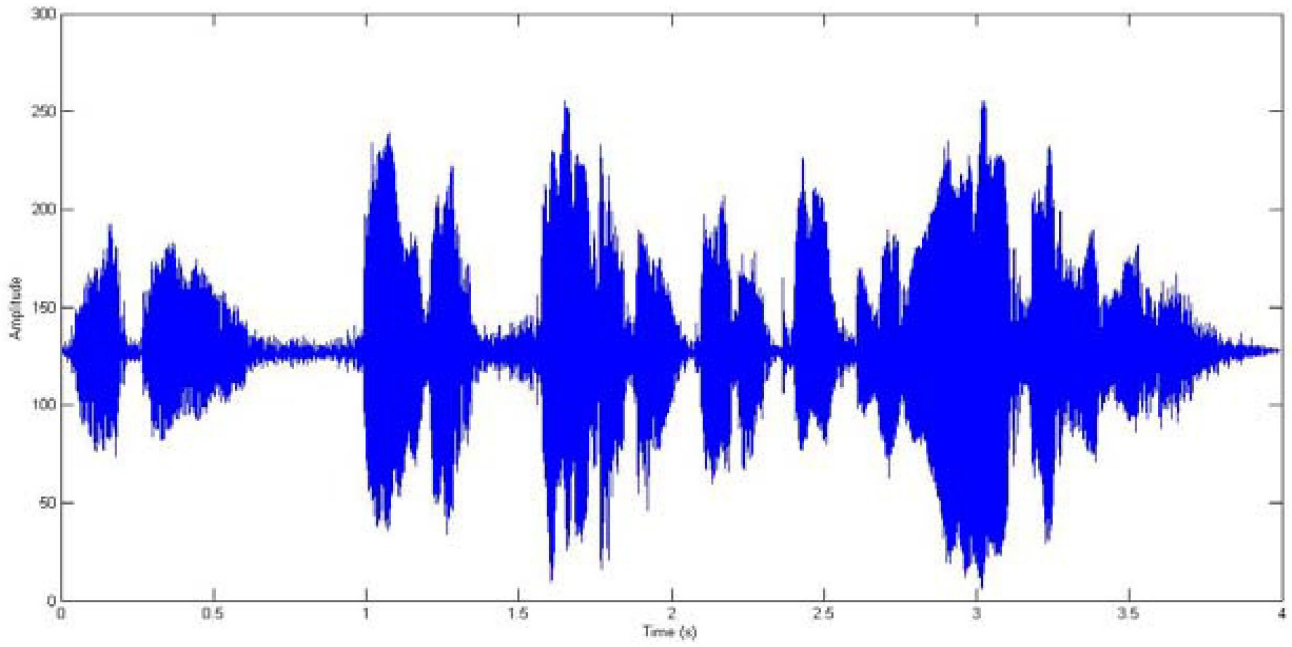
Figure 7. Encrypted File: *encrypted1.wav*

Here figure 6 and figure 10 shows two cover files. Figure 7-figure 9 and figure 11-figure 13 shows the corresponding encrypted file after embedding small, medium and large data sets into the cover file.



Medium Dataset: Message Length: 196

Figure 8. Encrypted File: *encrypted2.wav*



Large Dataset: Message Length: - 291

Figure 9. Encrypted File: *encrypted3.wav*

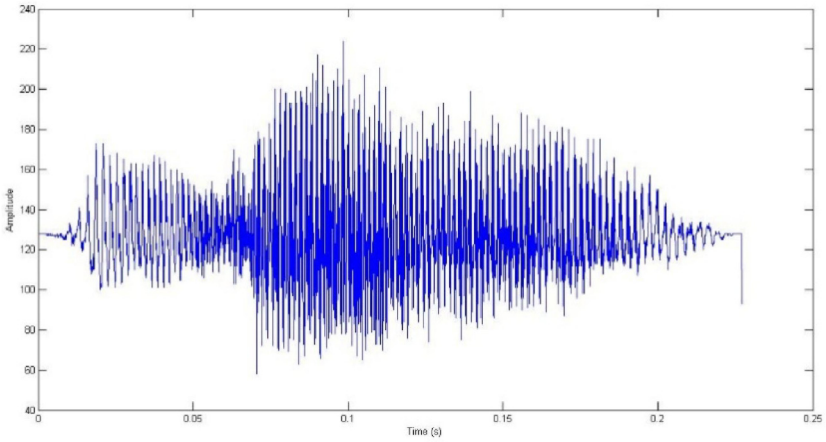
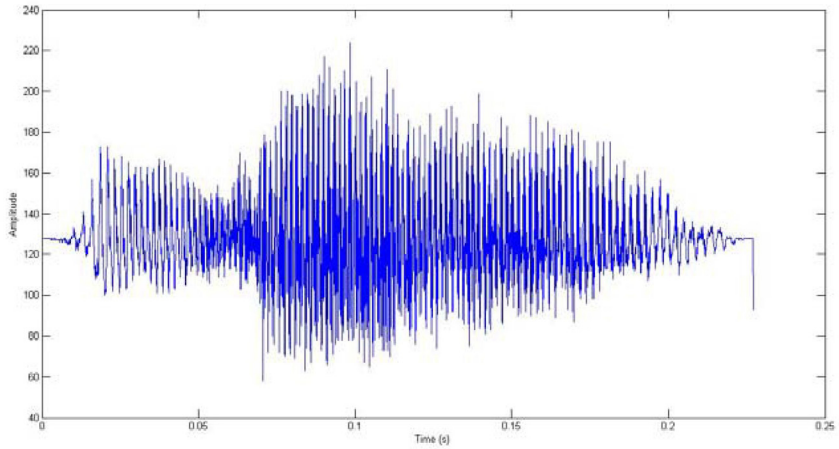
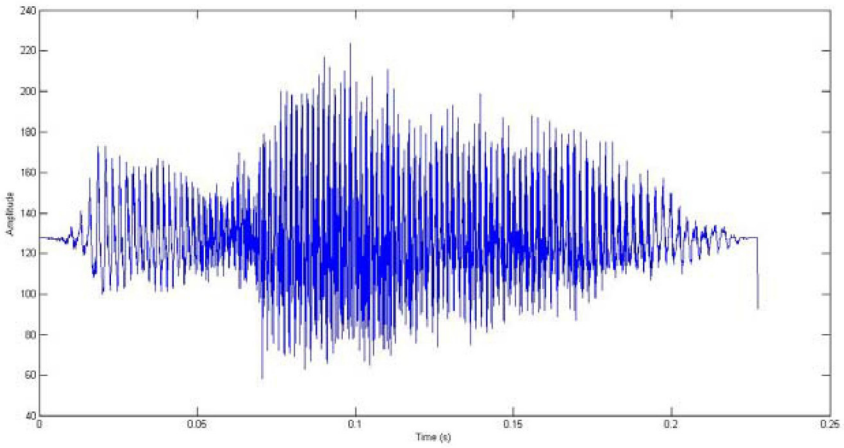


Figure 10. Cover file: Cover2.wav



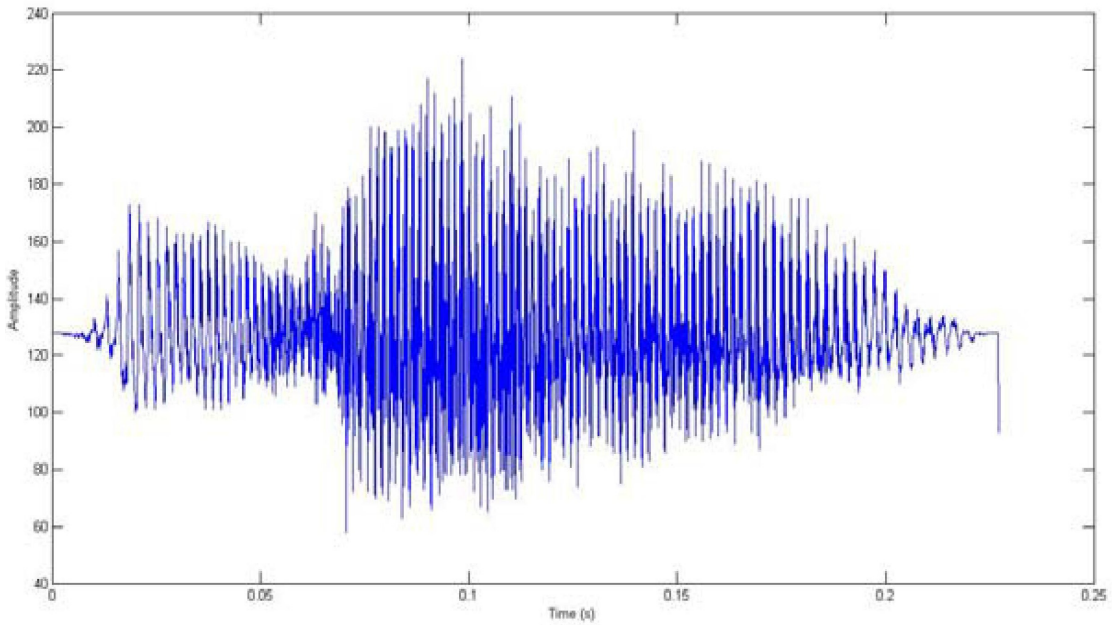
Small Dataset: Message Length 74

Figure 11. Encrypted File: *encrypted1.wav*



Medium Dataset: Message Length 202

Figure 12. Encrypted File: *encrypted2.wav*



Large Dataset: Message Length 549

Figure 13. Encrypted File: *encrypted3.wav*

From the above graph it is observed that original cover file and the corresponding stego file after embedding the message is reasonably same and their sound also audibly same .There is no way to identify the existence of the target string into the stego file. In the decryption end by applying reverse pattern matching algorithm the target string is decrypted from the stego file which is exactly same with the target string and its format is also unchanged. So it is concluded that it is a lossless pattern matching algorithm.

Now the qualities of the proposed method are analyzed by using two parameters Means square error calculation and Signal to noise ratio calculation and a comparison has been done with traditional LSB method shown in table 1.

Cover File	Length of Message	Normal LSB Method		Pattern Matching Method	
		MSER	SN Ratio	MSER	SN Ratio
1	50	0.0799	53.2623	0.1174	51.5891
1	100	0.1613	50.2088	0.2185	48.8927
1	200	0.3203	47.2309	0.4213	46.0401
2	70	0.0040	66.1247	0.0057	64.6388
2	140	0.0081	63.0987	0.0110	61.7356
2	250	0.0147	60.4891	0.0191	59.3635

Table 1. Comparison with respect to MSE and SNR

As compared to traditional LSB method we say that it is a common method and there is no security of data. The whole target string store into the LSB of the cover file without any encryption. So it is very easy for an intruder to identify the target string from the stego file. But our pattern matching method uses a different duel encryption methodology and for encryption purpose we have replaced the four bits sequence of the target string into five bits location number as a result the Means Square has slightly increased and SNR value has slightly decreased from LSB method. Although there is an increased in means square error compared with LSB method but 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 MSE and decreasing SNR.

7. Conclusion

In this paper a high quality duel encryption methodology has been implemented. A huge volume of target string is incorporated with then cover file using traditional LSB method and the target string has been accepted in the receiving side in a lossless manner. So it is concluded that the integrity and quality of the target string are well maintained. The experimental result have conformed this conclusion.

The proposed work 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. We can use our algorithm with traditional encryption to increase the level of encryption.

References

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

[2] Vimal, Jithu., Ann Mary Alex. (2014). Audio steganography using dual randomness LSB method. *In*: International Conference onControl, Instrumentation, Communication and Computational Technologies (ICCICCT), p. 941-944. IEEE, (Jul 10-11), Tamilnadu, India.

[3] Zamani, M., Manaf, A., Ahmad, R. B., Jaryani, F., Taherdoost, H., Zeki, A. M. (2009). A secure audio steganography approach, *In*: International Conference on Internet Technology and Secured Transactions, (ICITST) (November 9) p. 1-6. IEEE, London, UK.

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

[5] Din, Roshidi., Hanizan Shaker Hussain, Shuib, Sallehuddin. (2006). Hiding secret messages in images: suitability of different image file types, WSEAS TIONSRANSAC *on* COMPUTERS, vol. 6 (1) (January 1) 127 -132.

[6] 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.

[7] Rahim, L. B., Bhattacharjee, S., 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) 2014 Jan 1 (p. 277-289). Springer Singapore.

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

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

[10] Nathan, Mark., Parab, Nikhil., Talele, K. T. (2011). Audio Steganography Using Spectrum Manipulation. *In*: *T*echnology Systems and Management, p. 152-159. Springer Berlin Heidelberg.

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

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

[13] Bandyopadhyay, S. K., Bhattacharyya, D., Ganguly, D., Mukherjee, S., Das, P. (2008). A tutorial review on steganography. *In*: International Conference on Contemporary Computing 2008 Aug 7 (Vol. 101).

[14] Radhakrishnan, R., Kharrazi, M., Memon, N. (2005). Data masking: A new approach for steganography? *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology.* 2005 Nov 1. 41(3) 293-303.