Hiding Confidential File using Audio Steganography

Md. Rasedur Rahman
Department of CSE,
Jahangirnagar University,
Bangladesh

Partha Chakraborty
Department of CSE,
Comilla University,
Bangladesh

Md. Zahidur Rahman
Department of CSE,
Comilla University,
Bangladesh

Md. Golam Moazzam
Department of CSE,
Jahangirnagar University,
Bangladesh

ABSTRACT

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. In this research we have proposed a simple and effective way to hide data in audio file with 256 bit AES encryption based on Least Significant Bits technique. This approach can hide not only text message but also any type of files currently used e.g. .jpg, .pdf, .mp3, mpeg4 etc in a wav audio file.

Keywords

Cryptography, Decryption, Encryption, LSB, Secret key, Steganography, Stego-image

1. INTRODUCTION

A Hiding data and communications have existed since there is a need to communicate secretly with others. Steganography techniques are used to address digital copyrights management, protect information, and conceal secrets [1]. It is the art and science of hiding information into picture or other media in such a way that no-one apart from the sender and intended recipient even realizes there is a hidden information [2], [3]. This includes the concealment of digital information within computer files. Generally, a steganographic message will appear to be something else, may be picture, video, sound file, even the radio communication. This apparent message is the cover text. For instance, a message may be hidden by using invisible ink between the visible lines of innocuous documents. The hidden information is called stego-message which may be open message, but may be encrypted one as well. Data hiding [4] in the image has become an important technique for image authentication. Ownership verification and authentication is the major task for military people, research institute and scientist. It refers to the nearly invisible embedding of information within a host data set as message, image, and video. In steganographic [5], [6] applications, the hidden data may be secrete message or secrete hologram or secrete video whose mere presence within the host data set should be undetectable; The goal of steganography is to hide the message in the source image by some key techniques and cryptography is a process to hide the message content. To hide a message inside an image without changing its visible properties [7] the source image may be altered. The most common methods to make these alteration involves the usage of the least-significant bit (LSB) developed by masking, filtering and transformations on the source image [8].

This paper focuses on the technique to secure data or message with authenticity and integrity. The entire work has been done in WPF C#. In this work, the secret message is encrypted before the actual embedding process starts. It uses a simple encryption technique and a secret key and hence it will almost be impossible for the intruder to unhide the actual secret message from the embedded cover file without knowing secret key. Only receiver and sender know the secret key. N-bit LSB substitution technique is used as embedding and extraction method.

2. LITERATURE REVIEW

Sealed Information security and image authentication has become very important to protect digital image document from unauthorized access. Data is the backbone of today's communication. To ensure that data is secured and does not go to unintended destination, the concept of data hiding came up to protect a piece of information. Many researchers' have explored their innovative idea to implement steganography. The most famous method of traditional steganography technique around 440 B.C. is marking the document with invisible secret ink, like the juice of a lemon to hide information [9], [10].

Another method is to mark selected characters within a document by pinholes and to generate a pattern or signature [11]. Warkentin proposed an algorithm to hide data inside the audio visual files. Secret message will be hidden in a carrier file [12].

On the other hand, El-Emam [13], has proposed another algorithm to hide a huge amount of data that can be audio, image and text file inside of a color bitmap image where has maintained high security. He has filtered and segmented of image file instead of bits replacement is used on the appropriate pixels. Moreover, another researcher Chen has given concept to hide data inside of image edge portions that is modified method of El-Emam [14].

Rosziati Ibrahim and Teoh Suk Kuan [15] proposed an algorithm to hide message inside image. In the system data has been taken from input box as a text then converted it to binary codes. After converting, binary code is compressed as a file and then hide inside of Image. This steganography technique is used as an Image Steganography.

3. PROPOSED APPROACH

Least Significant Bit (LSB) method is good for audio steganography. LSB-Steganography is a steganography technique in which we hide messages inside an image by replacing least significant bit of career with the bits of message to be hidden. Fig. 1 shows the working logic of LSB method for audio steganography.

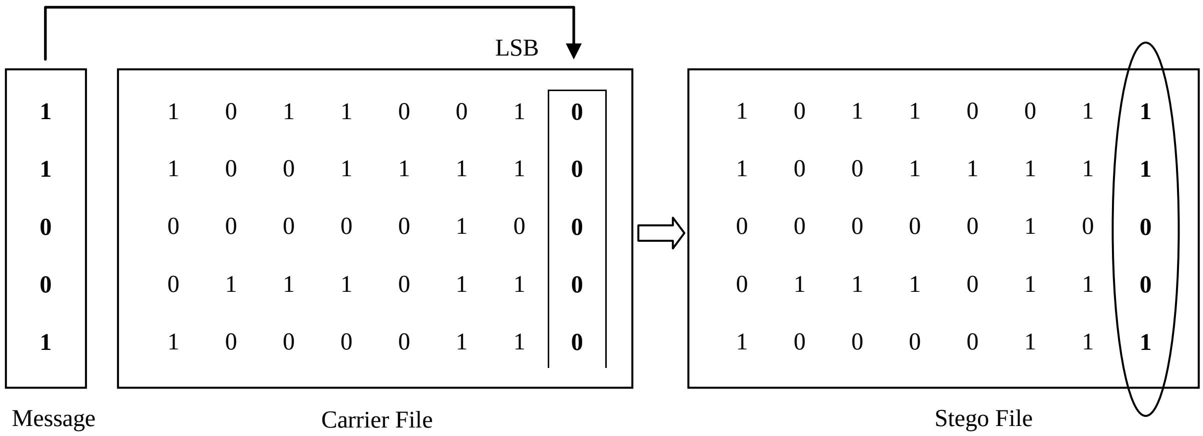


Fig. 1: LSB Method

Left 1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	**1**	**1**
Right 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**
Left 2	1	1	1	1	1	1	1	0	1	1	1	0	0	0	**1**	**1**
Right 2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	**1**	**1**

(a) Original Data Bytes

Left 1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	**0**	**1**
Right 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	**0**
Left 2	1	1	1	1	1	1	1	0	1	1	1	0	0	0	**0**	**0**
Right 2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	**1**	**0**

(b) Modified Data Bytes

Fig. 2: Modified LSB Method

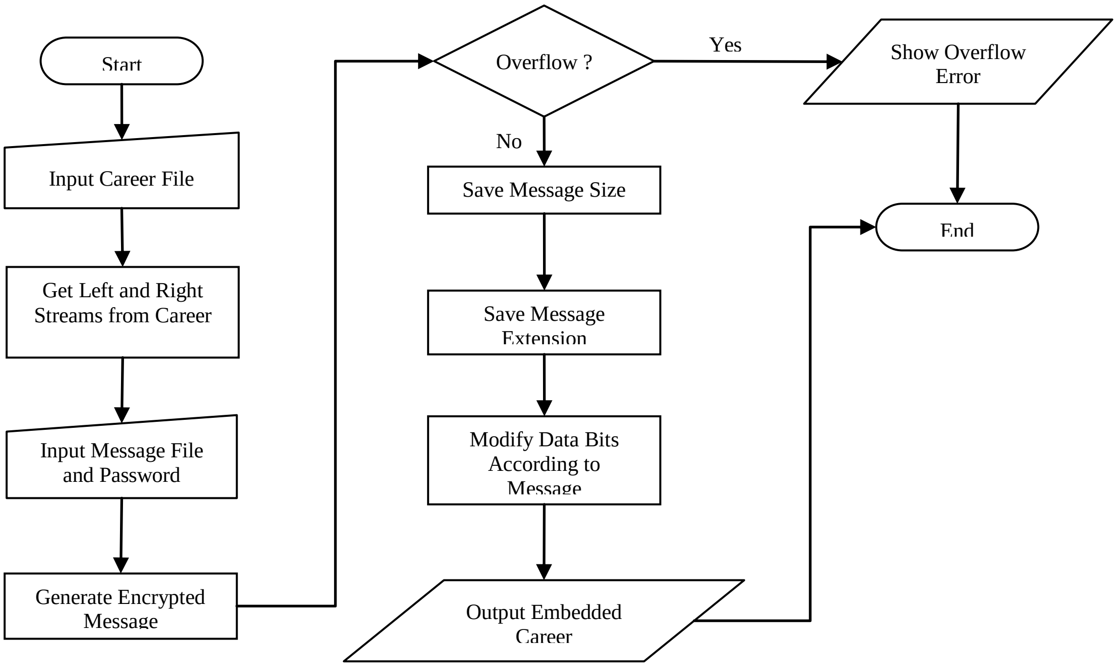

Start
Input Career File
Get Left and Right Streams from Career
Input Message File and Password
Generate Encrypted Message
Overflow ?
Yes
No
Show Overflow Error
Save Message Size
Save Message Extension
Modify Data Bits According to Message
Output Embedded Career
End

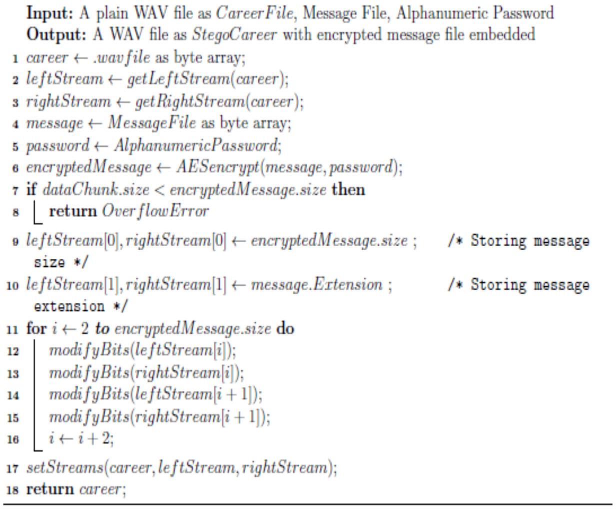
Fig. 3: Flowchart of Proposed Embedding Process in Audio Steganography

It is cleared from the WAV file structure that first 44 bytes of the file is format related data. These bytes should not be changed. If so, the file will be corrupted and no longer been usable. It can only temper data bytes starting from file offset 44. In the data subchunk, there is first two bytes is left channel data, next two bytes are right channel data and so on. It will change 2 bits of each two byte block of data as shown below in Fig. 2.

It is observed that when it increased the number of bits to change, it generates noise. When it decreased the bits to change, it's embedding capacity decreases. After some experiments, it decided to stick with 2 bits because it is the best combination of noise and embed capacity. Fig. 3 shows the pictorial representation of the proposed embedding process in audio steganography.

Proposed Embedding Algorithm in audio steganography is as follows:

Algorithm: Proposed Embedding Approach



```
Input: A plain WAV file as CareerFile, Message File, Alphanumeric Password
Output: A WAV file as StegoCareer with encrypted message file embedded
career ← .wavfile as byte array;
leftStream ← getLeftStream(career);
rightStream ← getRightStream(career);
message ← MessageFile as byte array;
password ← AlphanumericPassword;
encryptedMessage ← AESencrypt(message, password);
if dataChunk.size < encryptedMessage.size then
    return OverflowError
leftStream[0], rightStream[0] ← encryptedMessage.size ;      /* Storing message
    size */
leftStream[1], rightStream[1] ← message.Extension ;          /* Storing message
    extension */
for i ← 2 to encryptedMessage.size do
    modifyBits(leftStream[i]);
    modifyBits(rightStream[i]);
    modifyBits(leftStream[i + 1]);
    modifyBits(rightStream[i + 1]);
    i ← i + 2;
setStreams(career, leftStream, rightStream);
return career;
```

The data extraction procedure is the inverse of the data embedding procedure. In this procedure the secret message is extracted from the stego-file. The receiver inputs the stego-file to the data extraction algorithm.

The LSBs of each pixel of stego-file is extracted and placed in an array. This output is actually encrypted form of the original message. Then the message is decrypted using the same method that is used in encrypting. Fig. 4 shows the pictorial representation of the proposed extracting process in audio steganography.

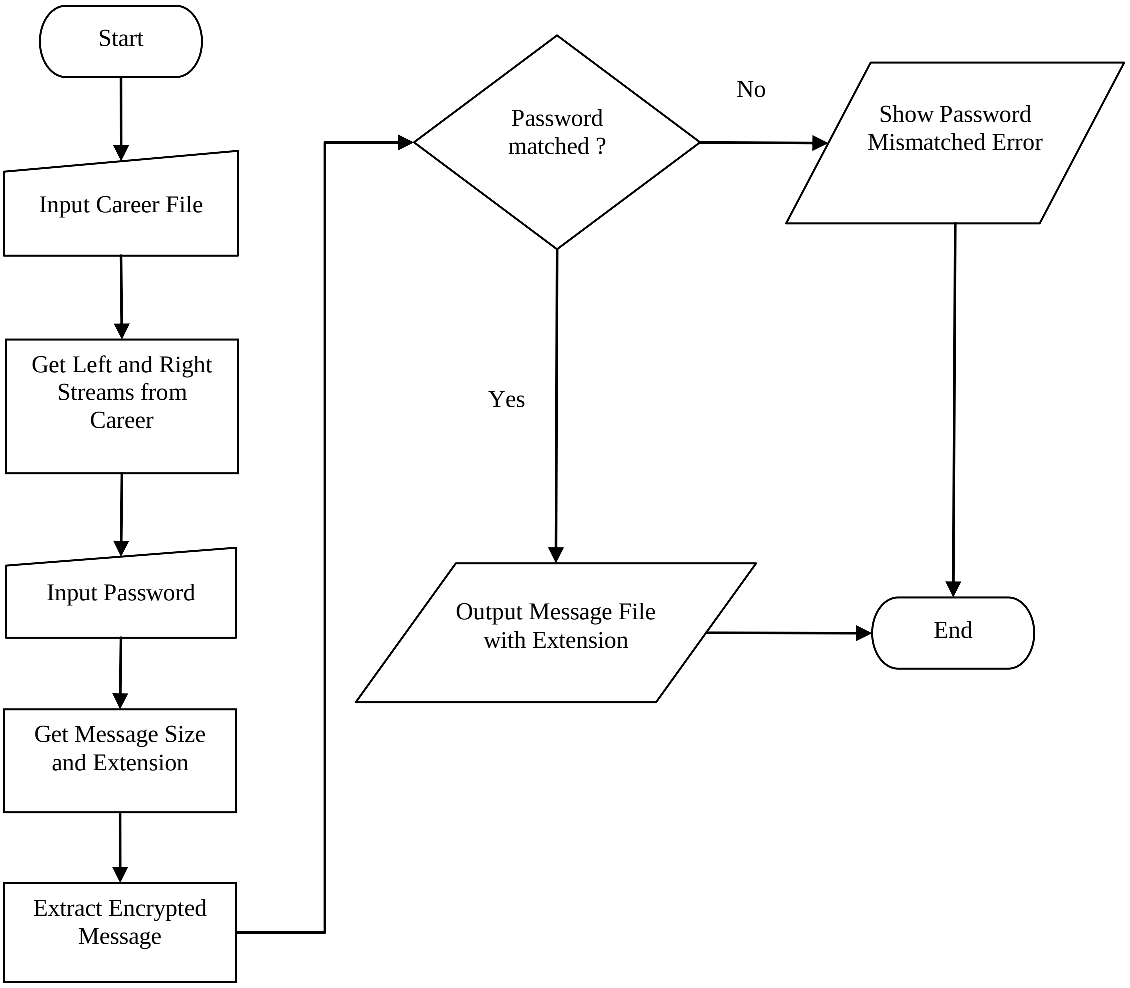
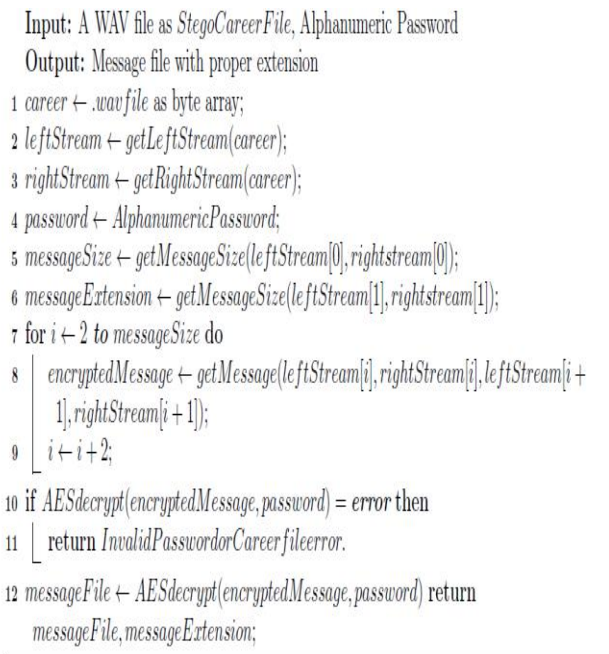

Start
Input Career File
Get Left and Right Streams from Career
Input Password
Get Message Size and Extension
Extract Encrypted Message
Password matched ?
No
Show Password Mismatched Error
Yes
Output Message File with Extension
End

Fig. 4: Flowchart of Proposed Extracting Process in Audio Steganography

Proposed Extracting Algorithm in audio steganography is as follows:

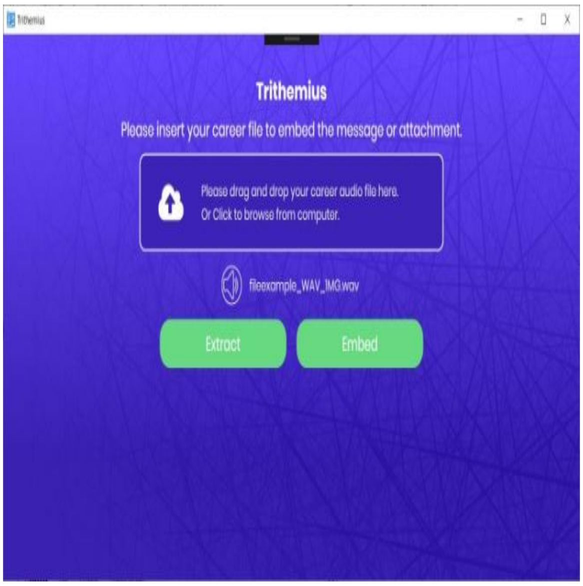
Algorithm: Proposed Extracting Approach



```
Input: A WAV file as $StegoCareerFile$, Alphanumeric Password
Output: Message file with proper extension
$career \leftarrow .wavfile$ as byte array;
$leftStream \leftarrow getLeftStream(career)$;
$rightStream \leftarrow getRightStream(career)$;
$password \leftarrow AlphanumericPassword$;
$messageSize \leftarrow getMessageSize(leftStream[0], rightstream[0])$;
$messageExtension \leftarrow getMessageSize(leftStream[1], rightstream[1])$;
for $i \leftarrow 2$ to $messageSize$ do
    $encryptedMessage \leftarrow getMessage(leftStream[i], rightStream[i], leftStream[i+1], rightStream[i+1])$;
    $i \leftarrow i + 2$;
if $AESdecrypt(encryptedMessage, password) = error$ then
    return $InvalidPasswordorCareerfileerror$.
$messageFile \leftarrow AESdecrypt(encryptedMessage, password)$ return $messageFile, messageExtension$;
```

4. IMPLEMENTATION AND RESULTS

To experiment the idea and approach a desktop application named Thithemius is created implementing the proposed approach. The application is build with WPF C#.



Trithemius
Trithemius
Please insert your career file to embed the message or attachment.
Please drag and drop your career audio file here.
Or Click to browse from computer.
fileexample_WAV_1MG.wav
Extract
Embed

Fig. 5: Upload Career Window

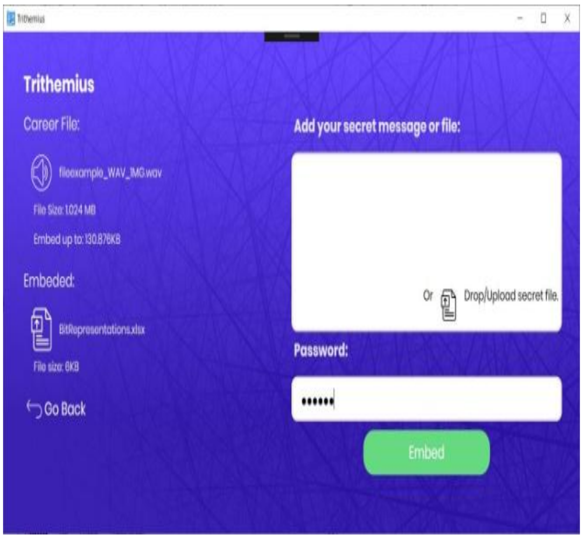


Fig. 6: Embed Window

The application has 4 main windows: **Upload Career Window**, **Embed Window**, **Extract Window** and **Save File Window**.

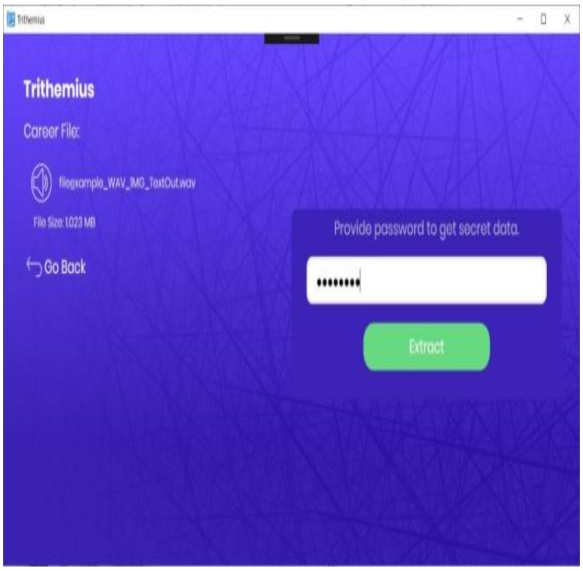
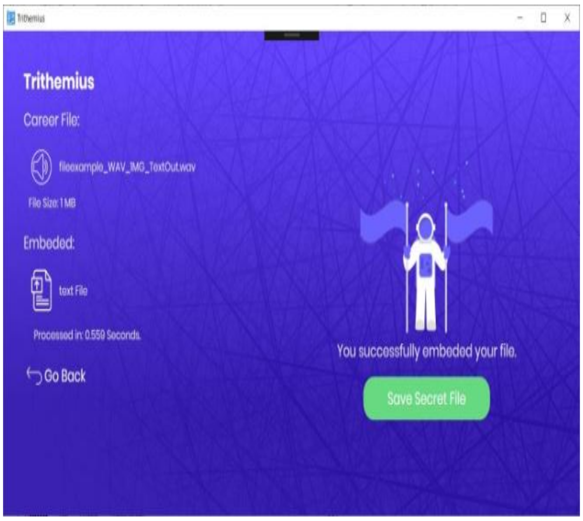


Fig. 7: Extract Window



Trithemius
Trithemius
Career File:
fileexample_WAV_1MG_TextOut.wav
File Size: 1 MB
Embeded:
text File
Processed in: 0.559 Seconds.
Go Back
You successfully embeded your file.
Save Secret File

Fig. 8: Extract Window

Table 1: Comparison of results

Career Size (MB)	Message Type	Message Size (MB)	Pass. Length (Char)	Embedded	Embed Time (Sec)	StegoCa-reer Size (MB)	Extracted	Extract Time (Sec)	Comment
1.049	Text	134017 Char	7	✓	0.851	1.048	✓	0.667	Successfull
9.925	.docx	0.069	8	✓	2.113	9.924	✓	1.017	Successfull
9.925	.ppt	0.209	8	✓	1.432	9.924	✓	1.160	Successfull
9.925	.pdf	0.999	8	✓	3.147	9.924	✓	2.263	Successfull
9.925	.jpg	0.802	8	✓	2.903	9.924	✓	2.160	Successfull
9.925	.zip	1.374	8	-	-	-	-	-	Unsuccessfull
9.925	.xlsx	0.006	8	✓	1.390	9.924	✓	1.347	Successfull
9.925	.wav	9.925	8	-	-	-	-	-	Unsuccessfull
9.925	.wav	4.985	8	-	-	-	-	-	Unsuccessfull
9.925	.json	0.001	8	✓	1.510	9.924	✓	1.364	Successfull
9.925	.txt	0.128	8	✓	1.779	9.924	✓	1.462	Successfull
4.985	.docx	0.069	8	✓	1.053	4.984	✓	0.874	Successfull
4.985	.ppt	0.209	8	✓	1.451	4.984	✓	1.213	Successfull
4.985	.pdf	0.999	8	-	-	-	-	-	Unsuccessfull
4.985	.jpg	0.802	8	-	-	-	-	-	Unsuccessfull
4.985	.xlsx	0.006	8	✓	1.013	4.984	✓	0.940	Successfull
4.985	.wav	9.925	8	-	-	-	-	-	Unsuccessfull
4.985	.json	0.001	8	✓	0.879	4.984	✓	0.975	Successfull
4.985	.txt	0.128	6	✓	1.603	4.984	✓	1.045	Successfull

Upload Career Window

In this window user can upload a WAV career file by file browser or drag and drop. File must be in .wav format, otherwise file can't be uploaded. From here user will go to Embed or Extract window.

Embed W indow

In this window user can embed message file within career file by file browser or drag and drop. User can also embed the text message. From here user will go to Save File window.

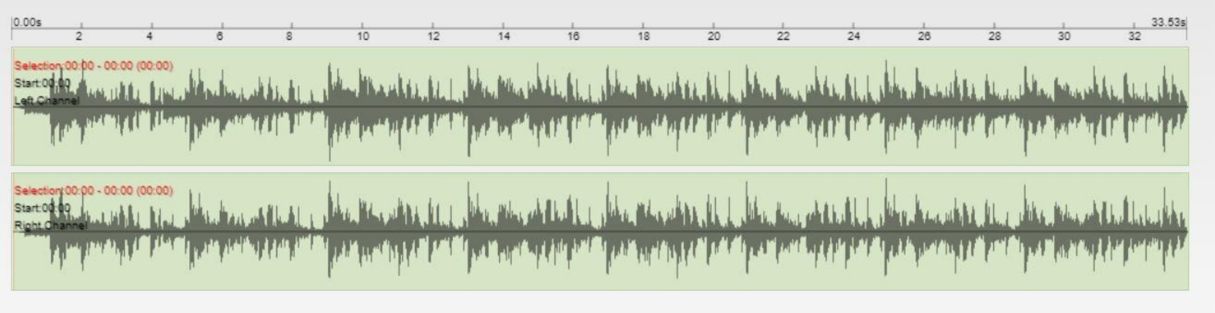


Fig. 9: Original inputted career signals

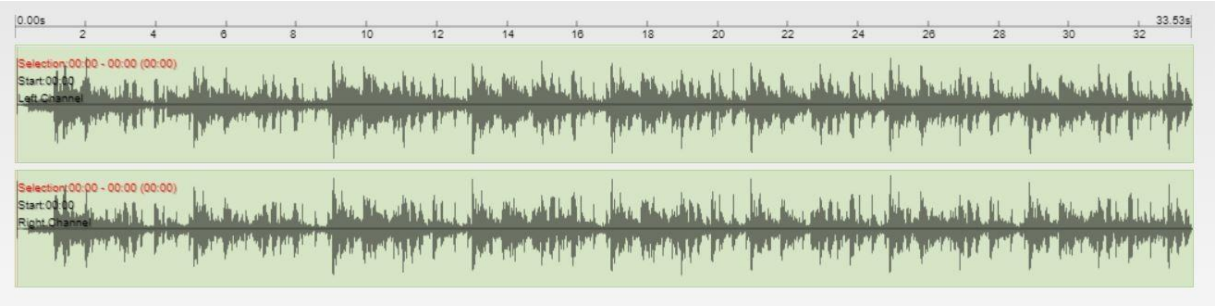


Fig. 10: Outputted stego-career signals

Extract Window

In this window user can extract message file from career file by providing correct password. From here user will go to Save File window.

Save File Window

In this window user can save the message file or Stego-career file and also shows the time of processing.

Comparison of Results

Using the application many different scenarios are tested. It embedded different file formats, different size of files, different passwords and then recovered it using the same application. A summary of testing is given in table 1.

Comparing Input and Output Career Spectrum

Using an audio editor it is found that there is merely any visible changes in both input and output career file. (Fig. 9 and Fig. 10).

According to the result and analyses it figured out unless capacity of career file is less than message file, every operation is successfully executed without any error.

5. CONCLUSION

This paper proposed a new approach of steganography with simpler implementation and good security. An application is developed using the approach to test and evaluate. In proposed approach it is found that the stego-career has not any noticeable noise on it. It also tested the output stego career size, because an unusual size of file can make doubt and this can cause compromise of hidden data. There is almost no change of file size. So, as a conclusion it could be stated that the new approach is efficient to hide the data.

6. REFERENCES

[1] Khare, P., Singh, J. and Tiwari, M., "Digital Image Steganography", Journal of Engineering Research and Studies, Vol. II, Issue III, pp. 101-104, 2011, ISSN: 0976-7916.

[2] Ghoshal N., Mandal, J. K., "Masking based Data Hiding and Image Authentication Technique (MDHIAT)", Proceedings of 16th International Conference of IEEE on Advanced Computing and Communications ADCOM-2008, ISBN: 978-1-4244-2962-2, December 14-17th, Anna University.

[3] S. Pavan, S. Gangadharpalli and V. Sridhar, "Multivariate entropy detector based hybrid image registration algorithm", IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Philadelphia, Pennsylvania, USA, pp. 18-23, March 2005.

[4] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information Hiding", IEEE Trans. On Info. Theory, Vol. 49, No. 3, pp. 563-593, March 2003.

[5] C. Rechberger, V. Rijman and N. Sklavos, "The NIST cryptographic Workshop on Hash Functions", IEEE Security & Privacy, Vol. 4, pp. 54-56, Austria, Jan-Feb 2006.

[6] C.Y. Lin and S. F. Chang, "A robust image authentication method surviving JPEG lossy compression", Proc. SPIE, Vol. 3312, San Jose, pp. 296-307, Jan. 1998.

[7] S. Dumitrescu, W. Xiaolin and Z. Wang, "Detection of LSB steganography via sample pair analysis", IEEE Trans. on Signal processing, Vol. 51, No. 7, pp. 1995-2007, 2003

[8] R., Chandramouli, and Nasir Memon., "Analysis of LSB based image steganography techniques." International Conference on Image Processing, 2001, IEEE, Vol. 3, pp. 1019-1022.

[9] Rabah, K., "Steganography – The Art of Hiding Data", Information Technology Journal, Vol.3, no.3, 2004, pp. 245-269.

[10] Curran, K. and Bailey, K., "An Evaluation of Image Based Steganography Methods", International Journal of Digital Evidence Fall 2003, Volume 2, Issue 2.

[11] Laskar, S.A. and Hemachandran, K., "An Analysis of Steganography and Steganalysis Techniques", Assam University Journal of Science and Technology, Vol.9, No.II, 2012, pp.83-103, ISSN: 0975-2773.

[12] M. Warkentin, M.B. Schmidt, E. Bekering, Steganography and steganalysis, Premier reference Source-Intellectual Property Protection for Multimedia Informaiton technology, Chapter XIX, 2008, pp. 374-380.

[13] N.N. El-Emam, Hiding a large amount of data with high security using steganography algorithm, Journal of Computer Science 3 (2007) 223-232.

[14] P.Y. Chen, W.E. Wu, A modified side match scheme for image steganography, International Journal of Applied Science & Engineering 7 (2009) 53-60.

[15] Rosziati Ibrahim, Teo Suk Kuan, Steganography Algorithm to Hide Secret Message inside an Image, Computer Technology and Application 2 (2011) 102-108.