Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Brief Study of Steganography on Different Cover Media's Using LSB Substitution Method

Prabhsimran Singh
Department of Computer Science and Engineering
Guru Nanak Dev University, Amritsar, India

Nitish Salwan
Department of Computer Science
Satyam Polytechnic & Pharmacy College, Amritsar , India

Sukhmanjit Kaur
Department of Computer Science and Engineering
Guru Nanak Dev University, Amritsar, India

Abstract— In today's world, security is the most eminent issue in field of communication. Lots of data security and data hiding algorithms have been developed in the last decade, which act as a base for modern security methods. This paper is going to discuss one of these methods know as steganography. Steganography aims to increase the security of data being sent over internet. This paper also discusses how the cover media/file is affected after the data is embedded in it using Least Significant Bit (LSB) substitution method.

Keywords— Data Hidding, Steganography, LSB, Stegofile and Cryptography.

I. INTRODUCTION

Steganography is the art of hiding of data within another file and the extraction of it at its destination. The word steganography is of Greek origin, where "Steganos" means "cover" and "Graphie" means "writing". Its ancient origins can be traced back to 440 BC. The Greek historian Herodotus writer of a nobleman, Histaeus, who used steganography first time[1]. Today in this modern world all the sensitive messages/information is being transferred online, it become very important to safely transfer this data. Steganography fully fill itself in this role, where the data to be send is hidden inside another file, so no one is able to know the presence of hidden data. Watermarking and Fingerprinting are two other technologies that are closely related to steganography, but they are mainly concerned with the protection of intellectual property[2]. Whereas steganography is concerned with the hiding of data in different files.

In this paper an effort is made to study data hiding in steganography using LSB substitution method on different cover media's such as Image, Audio and Video. Also checking how this data hiding effects the quality, sound, size and various other properties of the cover file. The implementation of the entire process is done in C# .NET.

II. IMPORTANT TERMONOLOGIES

A. ***Cover File:*** The file used for hiding data is called cover file and can be referred to as cover text, cover image, or cover audio as appropriate.

B. ***Stego File:*** After embedding the secret message it is referred to as stego medium or stego file.

C. ***Stego Key:*** A stego-key is used to control the hiding process so as to restrict detection and/or recovery of the embedded data.

III. PROCESS EXPLANATION OF STEGANOGRAPHY

The fig. 1 shows a simple representation of the encoding and decoding process in steganography. In this example, a secret data is being embedded inside a cover file(image in this case) to produce the stego image. A key is often needed in the encoding process. The encoding procedure is done by the sender by using the proper stego key. The recipient can extract the stego image in order to view the secret data by using the same key used by the sender.

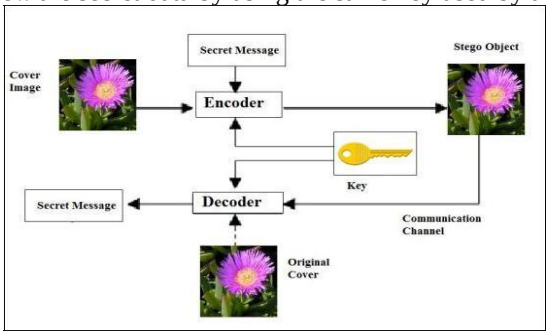


Fig. 1 "Block Diagram of Steganography Process"[3]

IV. ALGORITHM USED FOR ENCODING/DECODING

This Section discusses about the Least Significant Bit (LSB) algorithm used for encoding as well as decoding processes.

Least Significant Bit (LSB): Least Significant Bit (LSB) substitution method is a very popular way of embedding secret messages with simplicity[3]. The fundamental idea here is to insert the secret message in the least significant bits of the cover file. This actually works because the human visual system is not sensitive enough to pick out changes in color whereas changes in luminance are much better picked out. A basic algorithm for LSB substitution is to take the first N cover pixels where N is the total length of the secret message that is to be embedded in bits. After that every pixel's last bit will be replaced by one of the message bits.

Least significant bit (LSB) insertion is a common, simple approach for embedding information in a cover file. The LSB or in other words 8-th bit of some or all the bytes inside an image is changed to a bit of the secret message.

Let us consider a cover image contains the following bit patterns:

Byte-1	**Byte-2**	**Byte-3**	**Byte-4**
00101101	**00011100**	**11011100**	**10100110**
Byte-5	**Byte-6**	**Byte-7**	**Byte-8**
11000100	**00001100**	**11010010**	**10101101**

Suppose a number 200 is to embed in the above bit pattern. Now the binary representation of 200 is 11001000. To embed this information at least 8 bytes in cover file is needed. Now modify the LSB of each byte of the cover file by each of the bit of embed text 11001000.Now Table 1 shows what happens to cover file text after embedding 11001000 in the LSB of all 8 bytes.

Before Replacement	After Replacement	Bit Inserted	Remarks
00101101	00101101	1	No bit Replaced
00011100	**0001110*1***	1	Last bit changed
11011100	11011100	0	No bit Replaced
10100110	10100110	0	No bit Replaced
11000100	**1100010*1***	1	Last bit changed
00001100	00001100	0	No bit Replaced
11010010	11010010	0	No bit Replaced
10101101	**1010110*0***	0	Last bit changed

Table 1. "Bit replacement in LSB"

Here out of 8 bytes only 3 bytes get changed only at the LSB position. Since changing the LSB hence either changing the corresponding character in forward direction or in backward direction by only one unit and depending on the situation there may not be any change also as seen in the above example. As our eye is not very sensitive so therefore after embedding a secret message in a cover file our eye may not be able to find the difference between the original message and the message after inserting some secret text or message on to it [3].

V. IMPLEMENTATION

The implementation consists of Testing Software, Encoding Process and Decoding Process.

A. Testing Software

The software used to implement the working of Steganography is build in dot.net framework in Microsoft Visual C# 2010 Express edition, which is freeware provided by Microsoft[10]. The selection was purely based on simplicity and large pool of inbuilt functions and classes provided by dot.net which makes it quite easy to build this software. In this we simply try to embed group of words in various cover files and analyze the effect of this data hiding on cover files.

B. Encoding Process

ncoding is done at the sender site. First of all, a cover file in which the message or data to be send is going to be selected. Then a random secret/stego key is generated, which will be used by receiver to extract the hidden message at the time of decoding. Then using appropriate algorithm the cover file and message to be sent are converted into bytes, then by using the LSB(Least Significant Bit) substitution method the last bit i.e. the 8^{th} bit of every byte in cover file is replaced by the every bit of message. The File produced as a combination of cover file and hidden message is called stego file/object. This file has same size, color, quality and sound as of the original cover file. This makes is extremely difficult to detect the presence of hidden message. One important think that should be kept in mind is that the size of cover file should always be larger than the amount of data to be hidden in the cover file, else it would lead to corrupt stego file. The Fig. 2 shows the encoding interface of the software, where the message to be hidden is being embedded inside the selected cover file.

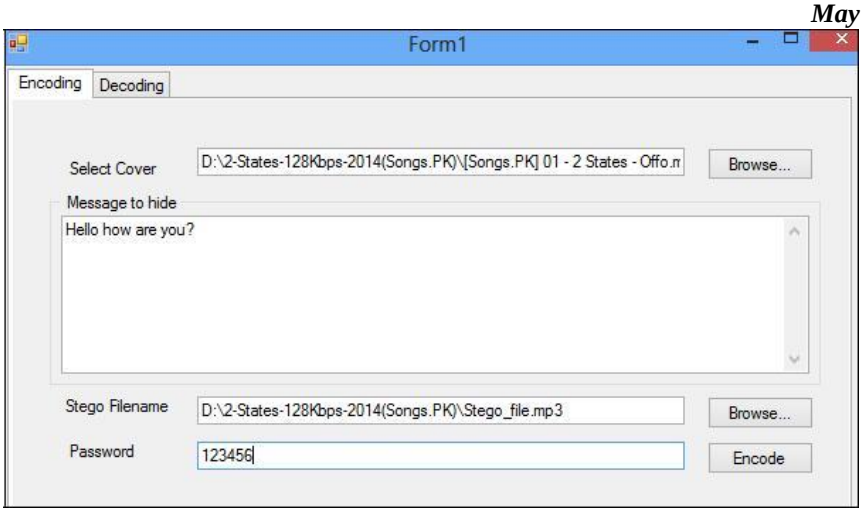


Fig. 2 "Encoding Interface"

C. Decoding Process

The process of decoding is performed at receiver side. The receiver uses the secret/stego key, generated by the sender. If the key is same as that was generated by the sender at time of encoding, the decoding process starts. In this the last bits of all bytes of stego file are extracted and combined together to form bytes of the hidden message. Then these bytes are converted to their original form i.e. the readable language. If the key is not the same then an unreadable form of message will be displayed to the receiver, this means even if the intruder somehow gets the stego file, if he doesn't have the secret/stego key, he/she will not be able to extract the hidden message. Fig. 3 shows the decoding interface and result produced if correct secret/stego key is used. While Fig. 4 shows decoding result with incorrect secret/stego key.

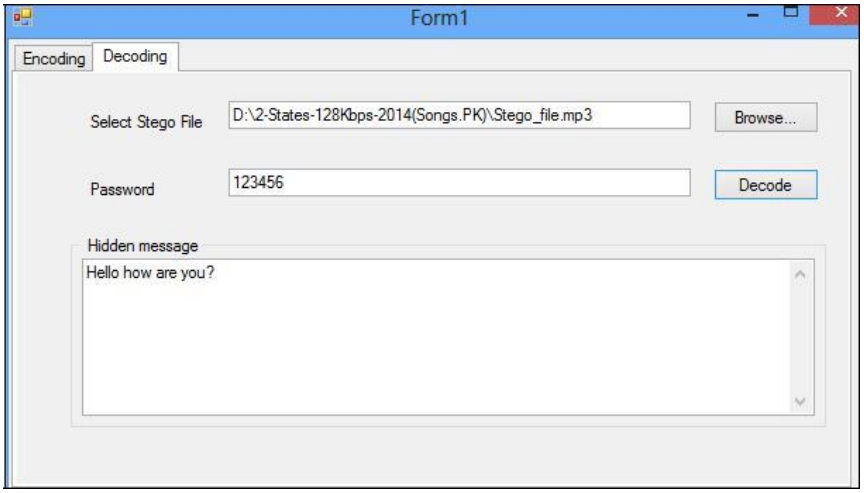


Fig. 3 "Decoding Result with correct secret/stego key"

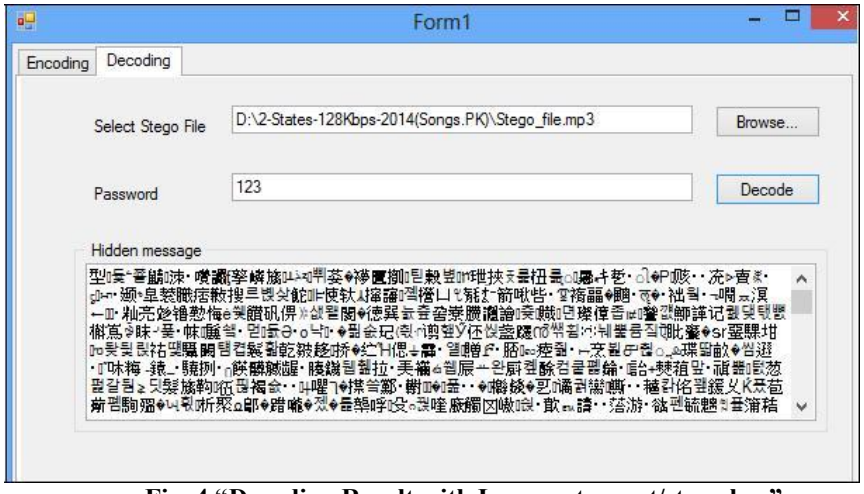


Fig. 4 "Decoding Result with Incorrect secret/stego key"

VI. ANALYSIS OF RESULTS

A. Image Steganography

In image steganography the cover file is an image file and can have an extension type .jpg, .png, .bmp etc. The aim in

this analysis is to compare the image quality and size of the original/cover file with stego file. Fig. 5 and fig. 6 shows image quality of both files while fig. 7 and fig. 8 shows the screenshot of properties of both the files.

Fig. 5 "Original File"

Fig. 6 "Stego File"

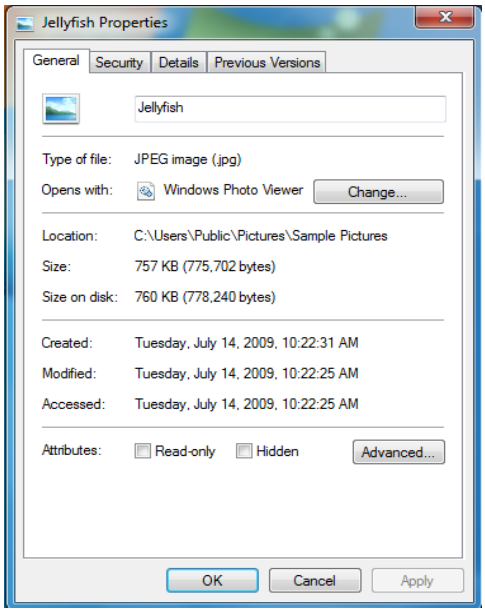


Fig. 7 "Properties of Original File"

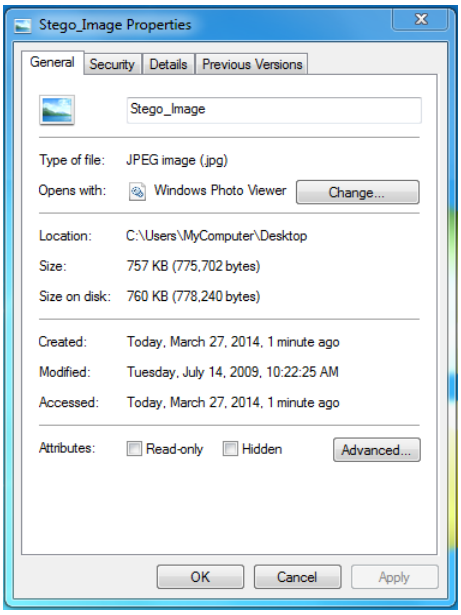
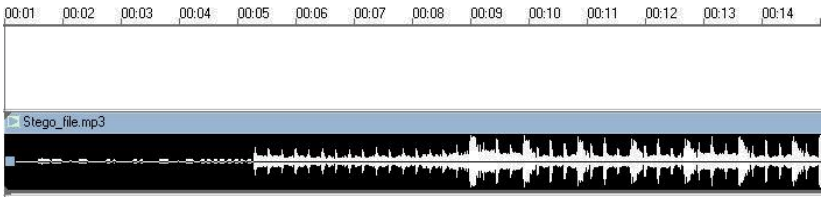


Fig 8 "Properties of Stego File"

B. Audio Steganography

In audio steganography the cover file is an audio file and can have an extension type .mp3, .wma, .m4a, .aac etc. The aim in this analysis is to compare the sound quality and size of the original/cover file with stego file. Fig. 9 and fig. 10 shows sound wave at a particular instant of both files while fig. 11 and fig. 12 shows the screenshot of properties of both the files.

Fig. 9 "Sound Wave of Original File"



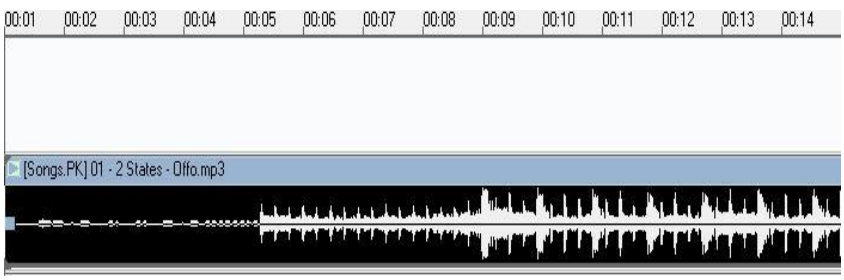


Fig. 10 "Sound Wave of Stego File"

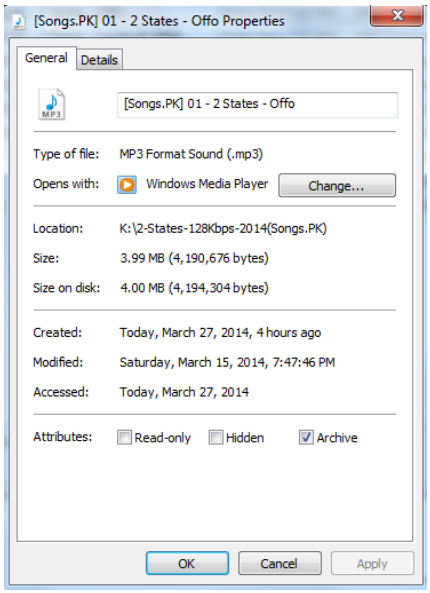


Fig. 11 "Properties of Original File"

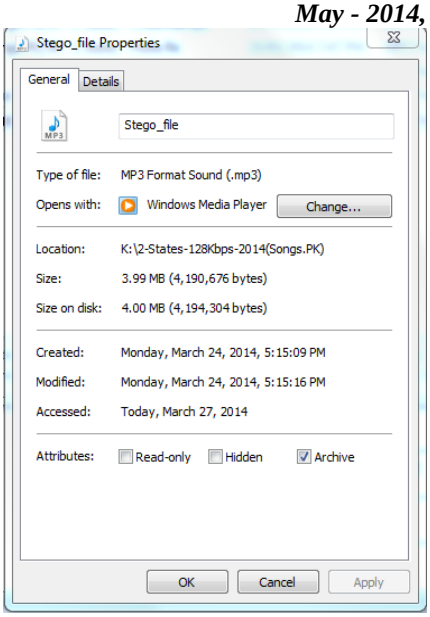


Fig. 12 "Properties of Stego File"

C. Video Steganography

In video steganography the cover file is a video file and can have an extension type .mp4, .avi, .mkv etc. The aim in this analysis is to compare the image, sound quality and size of the original/cover file with stego file. Fig. 13 and fig. 14 shows screenshot of video quality at a particular instant of both files while fig. 15 and fig. 16 shows the screenshot of properties of both the files.

Fig. 13 "Screenshot of Original file"

Fig. 14 "Screenshot of Stego file"

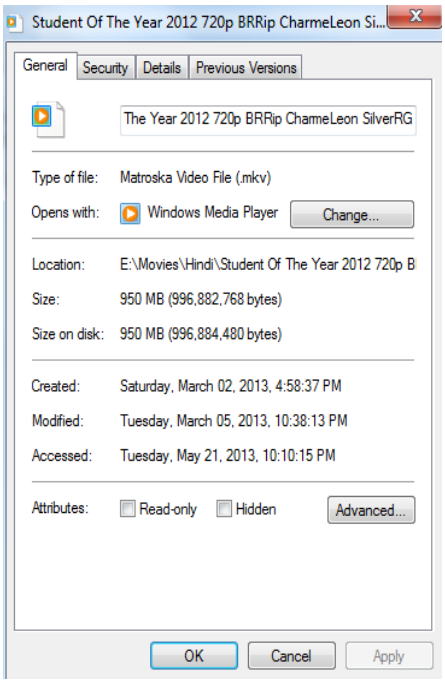


Fig. 11 "Properties of Original File"

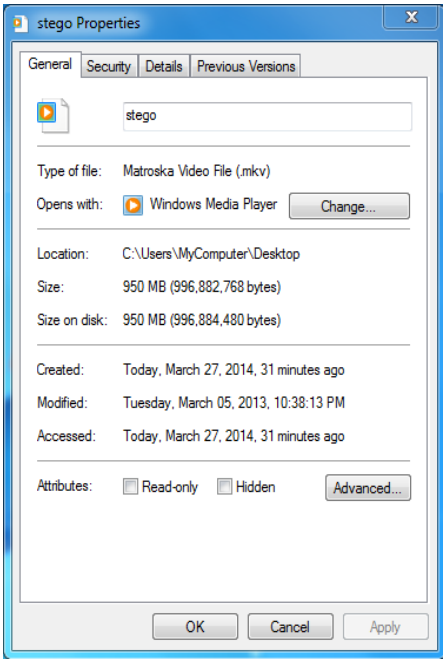


Fig. 12 "Properties of Stego File"

The main aim was to check whether the quality, size, sound and video after embedding the secret message in the cover file were affected or not. Since we are doing this with a purpose of transmitting secret message so that intruder may not know that the secret message is embedded inside that file particular cover file. If the quality, size, sound and video are not same in the stego file, this makes it clear for the intruder that there is something wrong with this file.

From the above results it is quite clear that data is successfully hidden in the cover file without effecting its size and quality, so it is impossible to detect difference between them. Hence the motive of successfully hiding data is being fulfilled using this technique.

VII. CONCLUSIONS

In this modern world, information hiding techniques are of great importance, earlier it was done with cryptography. Steganography has taken it a level up. It is not intended to replace cryptography but supplement it. In this paper we tried hiding the data in various cover files(Image, Audio and Video) and checked whether it affects their any property(quality, size, sound and video). Hence it was observed that using LSB substitution method the stego file obtained after hiding the secret message was exactly the same as the original cover file in all the test cases(Image, Audio and Video). So anyone scanning these files will fail to know that it contains hidden data.

REFERENCES

[1] Cambridge, UK, May30-June1 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.),pp.1-7 Benderr, D. Gruhl, N. Morimoto and A.Lu, "*Techniques for Data Hiding*", IBM Systems Journal, Volume 35, Issue 3 and 4, 1996, p.p., 313-336.

[2] Mr. Vikas Tyagi, "*Data Hiding in Image using least significant bit with cryptography*", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012.

[3] Harshitha K M, Dr. P. A. Vijaya, "*Secure Data Hiding Algorithm Using Encrypted Secret message*", International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012.

[4] L. M. Marvel, C. G. Boncelet, Jr. and C. T. Retter, "*Spread spectrum image steganography*," IEEE Trans. on Image Processing, 8(8), 1075-1083 (1999).

[5] R. Chandramouli, Nasir Memon, "Analysis of LSB Based Image Steganography Techniques", Proc. IEEE ICIP, 2001.

[6] Kevin Curran, Kran Bailey, "*An Evaluation of Image Based Steganography Methods*" International Journal of Digital Evidence,Fall 2003.

[7] G. Doërr and J.L. Dugelay, "A Guide Tour of Video Watermarking", Signal Processing: Image Communication, vol. 18, Issue 4, 2003, pp. 263-282.

[8] G. Doërr and J.L. Dugelay, "Security Pitfalls of Frameby-Frame Approaches to Video Watermarking", IEEE Transactions on Signal Processing, Supplement on Secure Media, vol. 52, Issue 10, 2004, pp. 2955-2964.

[9] K. Gopalan, "*Audio steganography using bit modification*", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal.

[10] http://www.visualstudio.com/en-S/products/visual-studio-express-vs

