

StegCure: A Comprehensive Steganographic Tool using Enhanced LSB Scheme

L.Y. Por¹, W. K. Lai², Z. Alireza³, T. F. Ang⁴, M.T. Su⁵, B. Delina⁶

Faculty of Computer Science and Information Technology

University of Malaya

50603, Kuala Lumpur

MALAYSIA

porlip@um.edu.my¹, waikit0513@gmail.com², zarrabi.alireza@gmail.com³, angtf@um.edu.my⁴,
smtng@um.edu.my⁵, delinabeh@yahoo.com⁶

Abstract: - Protected and encrypted data sent electronically is vulnerable to various attacks such as spyware and attempts in breaking and revealing the data. Thus, steganography was introduced to conceal a secret message into an unsuspecting cover medium so that it can be sent safely through a public communication channel. Suspicion becomes the significant key determinant in the field of steganography. In other words, an efficient steganographic algorithm will not cause any suspicion after the hidden data is embedded. This paper presents an overview of steganography on GIF image format in order to explore the potential of GIF in information hiding research. A platform, namely StegCure is proposed by using an amalgamation of three different Least Significant Bit (LSB) insertion algorithms that is able to perform steganographic methods. This paper explains about the enhancement of the Least Significant Bits (LSB) insertion techniques from the most basic and conventional 1 bit to the LSB colour cycle method. Various kinds of existing steganographic methods are discussed and some inherent problems are highlighted along with some issues on existing solutions. In comparison with the other data hiding applications, StegCure is a more comprehensive security utility where it offers user-friendly functionality with interactive graphic user interface and integrated navigation capabilities. Furthermore, in order to sustain a higher level of security, StegCure has implemented a Public Key Infrastructure (PKI) mechanism at both sender and receiver sites. With this feature, StegCure manages to restrict any unauthorized user from retrieving the secret message through trial and error. Besides, we also highlight a few aspects in LSB methods on image steganography. At the end of the paper, the evaluation results of the hybrid method in StegCure are presented. The future work will be focused in assimilation of more diversified methods into a whole gamut of steganography systems and its robustness towards steganalysis.

Key-Words: - steganography, GIF, security, information hiding, least significant bit, LSB.

1 Introduction

Steganography is derived from the Greek word steganos which literally means “covered” and graphia which means “writing”, i.e. covered writing [1], [18]. Currently, research in steganography has grown explosively in terms of further exploring message hiding within an object, a text or even a picture.

Steganography often draws confusion with cryptography [2] in terms of appearance and usage. The most significant difference between steganography and cryptography is the suspicion factor. In fact, when both cryptography and steganography are being implemented together, an acceptable amount of security could be achieved. Hence, a security utility which is called StegCure is built to essentially protect the privacy of confidential data with non-encryption method that is

without using direct password authentication during the transmission of information. This kind of method is used to make the presence of a secret data appear invisible to eavesdroppers such as keyloggers or harmful tracking cookies which can monitor a user’s keystroke when entering password and personal information. Information theft by malware has been widely used [24] to capture user’s password and confidential data in order to use it for hijacking personal possession especially counterfeiting or falsification of credit cards and identification cards. Therefore, StegCure can overcome this problem by embedding the data into GIF images so that it can be sent to the other party as an innocent looking file through the internet or a public domain during information exchange as in steganographic technique [3], [12]. Since the Internet is no longer a reliable means for sending

confidential messages, StegCure allows the information to be transmitted stealthily in contrast to conventional encryption techniques where the presence of encrypted information is exposed. This type of information may be very crucial to people such as unscrupulous business competitors or securities dealers or even those malicious group of organized crime figures, who may be keeping the sender or receiver under surveillance [28]. Therefore, it is essential to prevent the interception by other parties while transmitting data for safeguarding human to human communications.

StegCure was designed based on the conventional and general principle of steganography as illustrated in Figure 1. Basically, the secret data refers to a message which is saved as a text file that needs to be hidden. In this application, a GIF image will be chosen as a cover medium. The stego-image is the final product after a secret message is embedded in the cover object. Based on Figure 1, a secret message will be concealed in a cover-image by applying an embedding algorithm to produce a stego-image. The transmission of the stego-image via a communication channel is performed by a sender to a receiver. To reveal the covert message that is concealed by the sender, the receiver needs to have the de-stego algorithm which is parameterised by a stego-key to extract the secret message. This is the purpose of a steganographic system where an attacker who does not possess the name of a file or the stego-key for accessing it definitely will not be able to determine whether the file is even present [27]. In an efficient steganographic system, a normal cover medium should not be distinguishable from a stego-object [26].

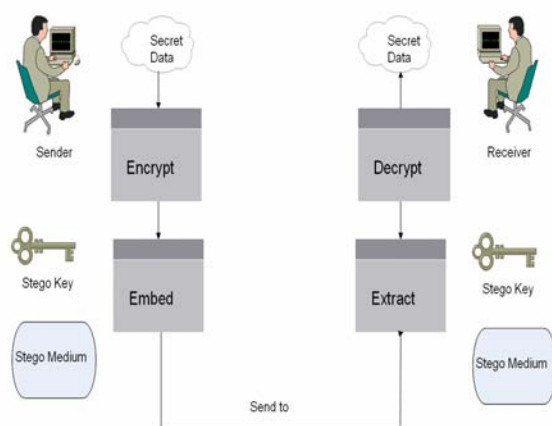


Fig.1: Steganography Mechanism

Digital images have become commonplace and nowhere are these images more prevalent than on the World Wide Web in the Internet [28]. Using digital images as a carrier medium is suitable for information hiding because of their insensitivity for the human visual system [11]. The vast majority of web pages are impressively sophisticated with colour images and thus Internet users browsing through the web no longer pay attention to sites containing images or to the downloading of images and data files from the Web [28]. Besides, there is a large amount of redundant bits in an image. The redundant bits of an object are those bits that can be altered but the alteration cannot be visibly detected by human eyes [13].

Based on the literature [10], [15] and [17], these existing LSB steganographic systems utilize only one LSB insertion method in concealing secret message. In addition, S-Tools and EzStego are two of the well-known steganography tools [17] that solely employ a single LSB method. However, the amalgamation of different steganography methods would enable the construction of a steganographic system that amasses the properties from various methods so that it provides a variety of algorithms for the user while increasing the difficulty of steganalysis at the same time. StegCure allows more hard codes to be added at the backend of the system by using function call to execute the method of additional algorithms.

2 Literature Review

This section reviews on the least significant bit (LSB) insertion method and the significance of using GIF format in StegCure.

2.1 Least Significant Bit insertion method

Least significant bit insertion is a common, simple approach to embed information in a cover file [6], [17]. The LSB is the lowest order bit in a binary value. This is an important concept in computer data storage and programming that applies to the order in which data are organized, stored or transmitted [21]. Usually, three bits from each pixel can be stored to hide an image in the LSBs of each byte of a 24-bit image. Consequently, LSB requires that only half of the bits in an image be changed [27] when data can be hidden in least and second least significant bits and yet the resulting stego-image which will be

displayed is indistinguishable to the cover image to the human visual system [17].

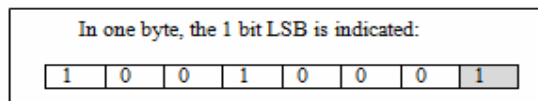


Fig. 2: Least Significant Bit

The last bit of the byte is selected as the least significant bit (as illustrated in Figure 2) because of the impact of the bit to the minimum degradation of images [12]. The last bit is also known as right-most bit, due to the convention in positional notation of writing less significant digit further to the right [7].

In bit addition (refer to Figure 3), the least significant bit has the useful property of changing rapidly if the number changes slightly. For example, if 1 (binary 00000001) is added to 3 (binary 00000011), the result will be 4 (binary 00000100) and three of the least significant bits will change (011 to 100).

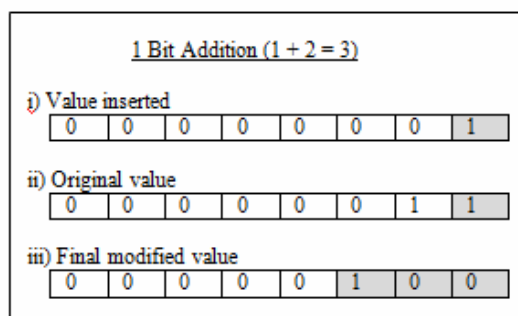


Fig. 3: Example of bit addition

Basically, by modifying the insignificant bits, the cover image is typically altered in a nearly imperceptible manner thereby ensuring that any observer would be unaware of the alteration made. Employing the LSB technique for data hiding achieves both invisibility and reasonably high storage payload, a maximum of one bit per pixel (bpp) for grayscale and three bpp for Red-Green-Blue (RGB) images [31].

There are a number of steganographic tools which employ LSB insertion methods available on the web. For example, S-Tools which is invented by Andy Brown, takes a different approach by closely approximating the cover image which may mean radical palette changes. S-Tools hides the secret message within the cover file via random available bits. These available bits are determined through the use of a pseudo-random number generator. Pseudo-

random is defined as random in appearance but reproducible by deterministic means, such as number generated by a series of equations. Once a pair is selected, the pixels intensities within one region are increased by a constant value while the pixels of the other region decreased by the same value [31]. The non-linear insertion makes the presence and extraction of secret messages more difficult. The image palette is taken and search for the LSB of each byte and the software then attempts to reconstruct the cover file by inserting the bits of the secret message into these LSBs [17]. Thus, S-Tools reduces the number of colours while maintaining the image quality, so that the LSB changes do not drastically change colour values. Based on the review [17], S-Tools provided the most impressive results of any steganographic package because S-Tools maintained remarkable image integrity.

Another tool, which uses LSB manipulation is EzStego written by Romana Machado. Based on Figure 4, EzStego arranges the palette to reduce the occurrence of adjacent index colours that contrast too much before it inserts the message. Two adjacent colours in the sorted palette can hardly be separated [5], [22]. The modification LSB in EzStego works by twiddling the least significant bit to encode the secret message. Then, it resorts the palette by renumbering all of the colours with their original value before shifting the image. The receiver resorts the palette using the same algorithm and extracts bits by using the sorted palette. This approach works quite well in gray-scale images and may also work well in images with related colours [17]. Apparently, based on literature [13], the problem with the palette approach used with GIF images is that when one changes the least significant bit of a pixel, it can result in a completely different colour since the index to the colour palette is changed [13]. Hence, StegCure manipulated the structure of the RGB component by enhancing the basic LSB method into a colour cycle algorithm so that the stego-image will not have drastic changes.

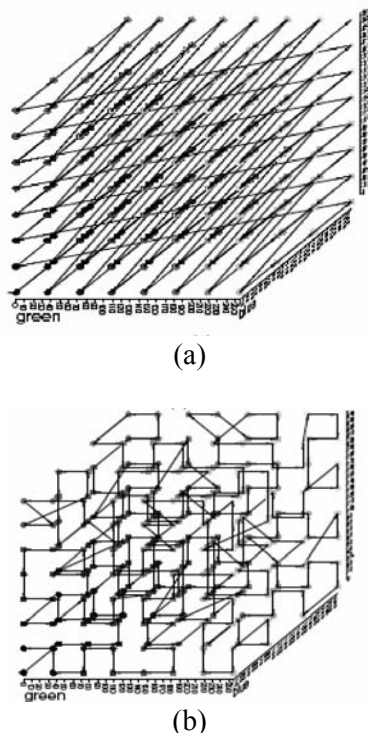


Fig. 4: A generic palette is shown on in (a) and the sorted version is shown in (b) (adapted from [22])

StegCure keeps the advantage of S-Tools and EzStego that is maintaining the image quality, but it can prevent the attack from hackers by restricting user to have only one attempt to perform destego. If the user has used the wrong destego method for the first time, there is no second attempt to recover the hidden data in the image even though the user then chooses the correct destego method.

Hide and Seek 4.1 by Colin Maroney is another basic steganography program that works on either 8-bit color or 8-bit black-and-white GIF files that are 320 by 480 [23] where this is the standard size of the oldest GIF format. The program displays the image before adding or extracting data. The current archive consists of two extra applications that provide feasibility for hiding information in GIF files which are grey.exe and reduce.exe. The former program converts colour GIFs into grayscale GIFs that would not show any of the artifacts associated with 8-bit colour steganography whereas the latter program shrinks the colour table from 256 colours to 128 colours and then duplicates these 128 colours so that adjacent entries in the colour table are duplicates of each other. Both of these applications can reduce the degradation of the image quality. In fact, there are some drawbacks especially when the

image is smaller than the minimum sizes (320x480), then the stego-image will be padded with black space. If the cover image is larger, the stego-image will be cropped to fit.

There is steganographic software named Outguess written by Niels Provos. The least significant bits are tweaked in a way to avoid introducing statistical signatures which may alert attackers looking for the presence of the message. Generally, OutGuess embeds message bits along a random walk into the LSBs of coefficients while skipping 0's and 1's. After embedding, the image is processed again using a second pass, but the corrections are made to the coefficients so that the stego image histogram matches the cover image histogram.

2.2 Least Significant Bit in GIF

Graphics Interchange Format, also known as GIF, is one of the machine independent compressed formats for storing images [14]. The significance of using GIF is because GIF is one of the most widely used image compression formats in web applications. LSB insertion in GIF is a very efficient algorithm to use when embedding a reasonable amount of data in a grayscale image [4],[13].

The compression scheme used in GIF is lossless which is called Lempel-Ziv-Welch (LZW). LZW reduces the file size without affecting the quality of the image [8]. LZW works by noting redundant areas of images, removing them and then reinserting them when the graphic is displayed [9].

GIF was developed by CompuServe to show images online (in the year 1987 for 8 bits video boards, before JPEG and 24 bits colour were used). Table 2 shows the bit depth and the number of colours in an image. The advantage of using GIF is it allows for a smaller storage file size and minimizes the transfer time over the network.

Table 2: Bit Depth and Numbers of Colour

24 bits	=	16777216 colours
16 bits	=	65536 colours
8 bits	=	256 colours
4 bits	=	16 colours

GIF images only have a bit depth of 8, the amount of information that can be hidden is relatively less than Windows Bitmap (BMP) [4]. According to the image analysis in [4], BMP is not widely used in web application [25] and thus the

suspicion might arise if it is transmitted with a LSB steganographic method. Heuristically, if more bits are altered it may result in a larger possibility that the degradation of the image can be detected with the human eye.

GIF images uses indexed colour, which contain a colour palette with up to 256 different colours out of 16,777,216 possible colours [32], and the Lempel-Ziv-Welch (LZW) compressed matrix of palette indices. Thus, LSB method in GIF is efficient when used for embedding a reasonable amount of data in an image [4].

Table 3: GIF Header Format (adapted from [14])

Offset	Length	Contents
0	3 bytes	"GIF"
3	3 bytes	"87a" or "89a"
6	2 bytes	Logical Screen Width
8	2 bytes	Logical Screen Height
10	1 byte	bit 0: Global Colour Table Flag (GCTF)
		bit 1..3: Colour Resolution
		bit 4: Sort Flag to Global Colourable
		bit 5..7: Size of Global Colour Table: $2^{(1+n)}$
11	1 byte	Background Colour Index
12	1 byte	Pixel Aspect Ratio
13	Not Fixed	Global Colour Table (0..255 x 3 bytes) if GCTF is one
	Not Fixed	Blocks
	1 bytes	Trailer (0x3b)

The composition of the file header (refer to Table 3) has to be identified to perform bit level manipulation on a GIF file so that it can avoid incorrect modification on the bit structure. Apparently, the magic number for GIF in file offset 0x00 is 0x47, 0x49, 0x46 and 0x38. In computer programming context [20], a magic number refers to a constant used to identify a file format. Its value or presence is inexplicable without some additional knowledge. The symbol of the hexadecimal representation is shown in Table 4.

Table 4: Magic number of GIF

Hexadecimal	0x47	0x49	0x46	0x38
Symbol	G	I	F	8

Manipulation of the bits in the image block could affect the colour schemes of the GIF image, but it

will not cause any distortion in the GIF image. The image block is the intended component in GIF image for steganography purpose. Table 5 shows the image block of GIF.

Table 5: GIF Image Block (adapted from [14])

Offset	Length	Contents
0	1 byte	Image Separator (0x2c)
1	2 bytes	Image Left Position
3	2 bytes	Image Top Position
5	2 bytes	Image Width
7	2 bytes	Image Height
8	1 byte	bit 0: Local Colour Table Flag (LCTF)
		bit 1: Interlace Flag
		bit 2: Sort Flag
		bit 2..3: Reserved
		bit 4..7: Size of Local Colour Table: $2^{(1+n)}$
	? bytes	Local Colour Table (0..255 x 3 bytes) if LCTF is one
	1 byte	LZW Minimum Code Size
Image block :		
[// Blocks		
1 byte Block Size (s)		
(s)bytes Image Data		
]*		
	1 byte	Block Terminator(0x00)

For example, if a user modifies a document file, he will not change the file format or the file properties in the bit level. The main purpose of implementation of steganography technology on GIF is to conceal the secret message into the colour bytes that could draw less suspicion. Therefore, it is not necessary to modify the file properties of the GIF or even corrupt the bit structure of the GIF file.

Subsequently, LSB method affects the minimum pixel value in the secret text embedding mechanism of the GIF image. It is generally assumed with good reason that the degradation caused by this embedding process would be perceptually transparent due to the weaknesses of human visual system [19].

3 StegCure Design

StegCure is proposed in image steganography which marries three different steganography algorithms in one single steganography application. The graphic user interface of StegCure is shown in Figure 5 and Figure 6. Eventually, the original image and the stego-image are shown in Figure 7 after performing steganography in data embedding.

The contribution of StegCure is that StegCure has implemented a PKI mechanism at both sender and receiver sites. PKI is a comprehensive system required to provide public-key encryption and digital signature services where its arrangement enable computer users without prior contact to be authenticated to each other public key certificates to encrypt messages to each other [33]. With this feature, StegCure manages to restrict any unauthorized user from retrieving the secret message through trial and error although the secret image has been intercepted by hackers or crackers in a communication channel.

The first algorithm that is included in the StegCure is the Least Significant Bit Steganography algorithm. Based on the algorithm in StegCure, the last byte is altered to store the information as shown in Figure 8.

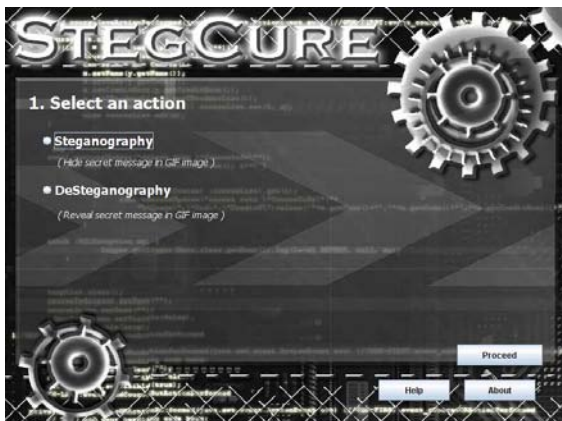


Fig.5: GUI for amalgamation of three steganography methods in StegCure



Fig.6: The steganography process in StegCure



Fig.7: The results of the steganography process

The second algorithm in StegCure is the modified Least Significant 1 Bit Steganography algorithm, i.e. Least Significant 2 Bits Steganography algorithm. In StegCure, the blue colour is considered as least significant byte among the other two colour bytes. In Figure 9, the right most two bits are selected as the interest bit for LSB insertion. Two most right bits of one pixel are selected as interest bit for substitution with two corresponding bits of the payload [10].

Red	Green	Blue
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

Fig.8: RGB structure in GIF pixel for Least Significant 1 Bit Algorithm

Red	Green	Blue
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

Fig. 9: RGB structure in GIF pixel for Least Significant 2 Bits Algorithm

As illustrated in Figure 8 and Figure 9, the changes of LSB cannot be detected because of the imperfect sensitivity of the human eyes [29]. It seems that the human eyes are less sensitive to blue colours among the 3 colours of RGB. Based on the optical research by Hecht in [30], the visual perception of intensely blue objects is less distinct than the perception of objects of red and green. By applying this concept into StegCure, more significant changes can be applied to blue colour. Therefore, in the second algorithm, the last two bits of blue colour are chosen as the least significant bit of the blue colour.

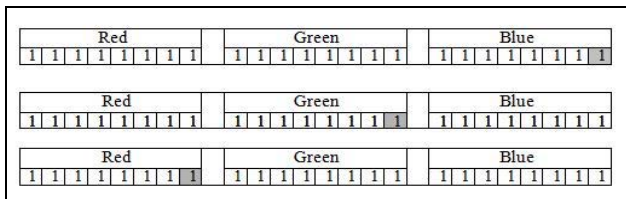


Fig. 10: RGB structure in GIF pixel for Least Significant Colour Cycle Algorithm

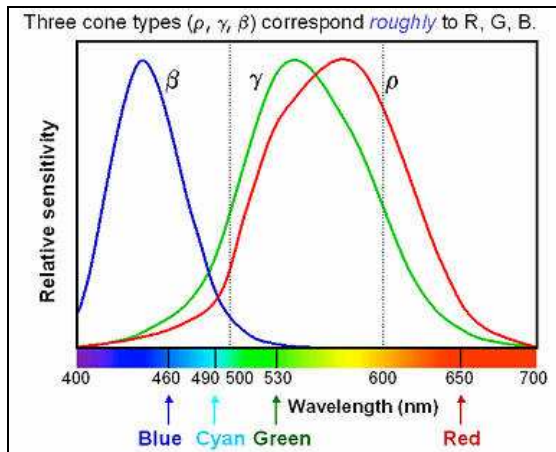


Fig.11: Human spectral sensitivity to colour (adapted from [16])

According to the optical research in [16], the retina, which is located at the back of the eye, contains three types of cones, each responds differently to light of various wavelengths. Based on Figure 11, LSB from each peak of the absorption (RGB) is selected to become the interest bit for information embedding in colour cycle so that gradient deviation is distributed evenly at 440nm, 550 nm, and 580 nm respectively.

The third algorithm in StegCure is the colour cycle (refer to Figure 10). The colour cycle algorithm involves a rotation of the bit substitution range from blue, green and red. Consideration on the Least Significant Byte concept is ignored in this colour cycle algorithm. The information embedded in the right most bit of each colour byte respectively. Each round of bit replacement only substitute one bit. The capacity of information hiding is similar to Least Significant 1 bit. The system flow of StegCure is illustrated in Figure 12.

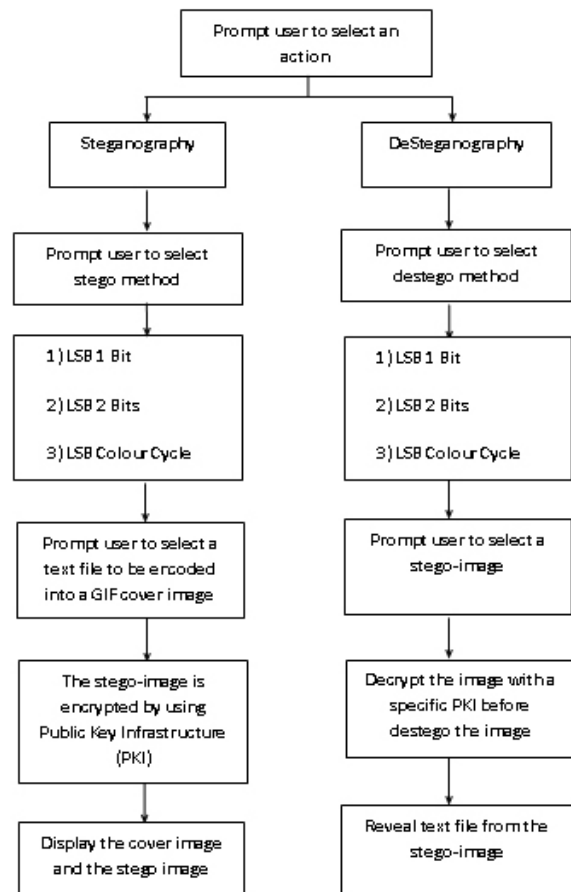


Fig.12: System flow of StegCure

StegCure is a user-friendly application where there is a real-time user guide to provide users a step-by-step instruction on using StegCure (as shown in Figure 5). In the main menu of StegCure, there are stego and destego functions. If a user wants to embed data and chooses stego, it will prompt user to select any of the three algorithms which are LSB 1 Bit, LSB 2 Bits or Colour Cycle LSB. Next, the user is required to browse the secret text file and select a GIF image as a cover medium. Then, StegCure will embed the text file into the selected cover image and then encrypt it with a stego-key. Finally both cover image and stego-image are displayed (as shown in Figure 7). The secret file will only be decrypted when the receiver chooses the correct stego-key and algorithm which are same as the sender in order to prevent interception from other parties.

4 System Testing and Evaluation

StegCure was developed using one of the most popular open source development technologies

which is JAVA programming language. This platform provides support on the construction of the entire system. After the completion of the system development, StegCure was tested using a personal desktop computer which is Intel Core 2 Duo Processor at 2.30 GHz along with 2048MB DDR RAM and also tested with a notebook using Intel Centrino Core 2 Duo 2.00 GHz built with 1024MB DDR RAM. It was found that StegCure can be run smoothly on both types of computers in JAVA environment.

Apparently, this system testing is conducted based on the functional requirements in the project. The objective of this test is to particularly evaluate the performance of the system whether the application can fulfil the system requirements as stated. The following table illustrates the percentage of the system evaluation based on 30 samples of users from the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia on its functionality and features.

Table 6: System evaluation based on thirty samples of users.

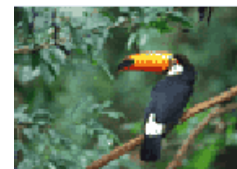
Functionality and Features	Evaluation (%)
Navigation flow	92.6
System feasibility, accessibility and user friendliness	78.1
Successful encoding	100.0
Successful decoding	100.0
Accuracy of information retrieval	100.0
Pleasant graphic user interface	79.8
Sufficient of cover payload	82.6
Imperceptibility of stego image	100.0
Effectiveness of hiding information	86.7
Applicability in real-life routine	96.4

Based on the system evaluation (as shown in Table 6) from the samples of users during the system testing, the encoding and decoding processes can work fantastically well to hide and reveal information. Furthermore, the secret information can be retrieved without encountering any loss of data. Most importantly, the modification of the cover image is not perceptible on the stego image at all and thus arouses no suspicion to third parties. A large number of users consider StegCure applicable in real-life routine like sending emails and confidential data over the Internet. There is a high percentage which indicates that the system has a smooth navigation where there is no broken links to go forth or backward during the steganographic process. According to the statistic, most of the users think that the capacity of the cover image and the effectiveness of information hiding are limited as it

has to be in GIF format. However the above mentioned issues have been resolved after justifying to the users about the fact of using GIF format can be easily unnoticeable to others because the file size is small and widely used in web application.

5 Preliminary Experimental Results

Relatively, the result of the stego-image (refer to Figure 13) does not generate any suspicion at all. The difference of the stego-image can hardly be distinguished after using the LSB method insertion. It is proven that human visual system is not able to differentiate the original image and the stego-image, but computer system can detect the modification of the bits through hexadecimal representation by using a hex editor tool. Figure 14 and Figure 15 illustrates the error messages when a user has selected some inappropriate options in StegCure.



Original Image


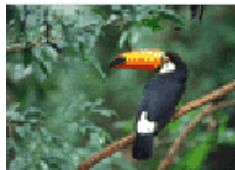

	LSB 1 Bit
	LSB 2 Bits
	LSB Colour Cycle

Fig.13: The results after the message embedding using StegCure



Fig.14: Error message when the data file is bigger than the cover

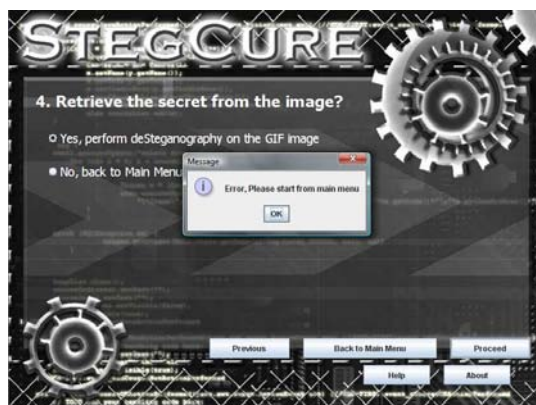


Fig.15: Error message of incompatible cover image

6 Conclusions and Future Work

A combination of three steganography algorithms on GIF image is proposed through StegCure system. The unique feature about the StegCure is being able to integrate three algorithms in one steganography system. By implementing Public Key Infrastructure, unauthorized user is forbidden from intercepting the transmission of the covert data during a communication because the stego-key is only known by the sender and the receiver.

As has been discussed in the related work, all of the foregoing methods suffer from various drawbacks. The future work should be focused towards exploring and developing steganography algorithms that will be more reliable so that they are able to hide significant size in cover image while maintaining the quality of the stego-image. The steganographic algorithm can also be able to sustain the level of robustness against steganalysis. Besides, in order to increase the efficiency to prevent hackers from downloading the stego-image more than one time and intercepting the secret data, a database can

be created to store both MAC address and IP address of the users. This is because MAC address and IP address are unique identifiers for computers and any other device on a TCP/IP network. Therefore, if the MAC or the IP address is detected occurring at the second time, user will be automatically barred from downloading the file. When this method works together with the PKI, StegCure can provide an acceptable amount of security and privacy during data transmission.

7 Acknowledgements

We would like to express our gratitude to Dr. Goh Chong Tien for proof reading and giving us feedback on the paper.

References:

- [1] Mohammed Al-Mualla and Hussain Al-Ahmad, "Information Hiding: steganography and Watermarking". [Online]. Available: http://www.emirates.org/ieee/information_hiding.pdf, [Accessed: March 12, 2008].
- [2] Alain Brainos, "A Study of steganography and the Art of Hiding Information", *SecurityWriter*, July 27, 2004.
- [3] Muthiyalu Jothir, Navaneetha Krishnan, "Statistical models for Secure steganography Systems", *Digital Rights Management Seminar*, 15th May, 2006.
- [4] Neeta Deshpande, Snehal Kamalapur and Jacobs Daisy, "Implementation of LSB steganography and Its Evaluation for Various Bits", *1st International Conference on Digital Information Management*, 6 Dec. 2006 pp. 173-178.
- [5] Westfield Andreas and Andreas Pfitzmann, "Attacks on Steganographic Systems", *Proceedings of Third International Workshop Computer Science IH'99 Germany*, 1999, pp. 61-76.
- [6] Chandramouli R and Memon N, "Analysis of LSB based image steganography techniques", *Proceedings 2001 International Conference on Image*, Vol. 3, pp. 1019-1022.
- [7] John Kadvany, "Positional Value and Linguistic Recursion", *Springer Netherlands*, Vol. 35, December 2007, pp. 487-520.
- [8] Memon, N. and Rodila, R. , "Transcoding GIF images to JPEG-LS", *Consumer Electronics, IEEE Transactions on Consumer Electronics*, Vol. 43, Issue 3, Aug 1997, pp. 423-429.

- [9] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding", *IEEE Transactions on Information Theory*, 1978, pp. 530-536.
- [10] Andrew D. Ker, "Steganalysis of Embedding in Two Least-Significant Bits", *IEEE Transactions On Information Forensics And Security*, Vol. 2, No 1, March 2007, pp. 46-54.
- [11] Der-Chyuan Lou and Jiang-Lung Liu, "Steganographic Method for Secure Communications", *Elsevier Science Ltd*, Vol 21, No 5, 2002, pp 449-460.
- [12] S. Katzenbeisser, Fabien A.P. Petitcolas, *Information Techniques for Steganography and Digital Watermarking*, Artech House, Boston, London, 2000.
- [13] T. Morkel, J.H.P. Eloff, M.S. Olivier, "An Overview of Image Steganography", *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, South Africa, 2005.
- [14] Graphic Interchange Format. [Online]. Available: <http://www.onicos.com/staff/iz/formats/gif.html>, [Accessed: April 18, 2008]
- [15] Mehmet Utku Celik, Gaurav Sharma, Ahmet Murat Tekalp and Eli Saber. "Lossless Generalized-LSB Data Embedding", *IEEE Transaction on Image Processing*, Vol. 14, No. 2, Feb 2005, pp. 253-266.
- [16] Y. S. Huang, Y. P. Huang, Ke-Nung Huang, M. S. Young, "The Assessment System of Human Visual Spectral Sensitivity Curve by Frequency Modulated Light", *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, September 2005, pp. 263-265.
- [17] N. F. Johnson, S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, February 1998, pp.26-34.
- [18] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn. "Information Hiding – A Survey", *Proceedings of the IEEE, special issue on protection of multimedia content*, July 1999, pp. 1062-1078.
- [19] Professor R. J. Solomon, "Weaknesses of Machine Vision", 27 Jan 2004. [Online]. Available: <http://cordis.europa.eu/ictresults/index.cfm/section/news/tpl/article/BrowsingType/Features/ID/60585>, [Accessed: April 24, 2008]
- [20] Magic number, [Online]. Available: <http://www.economicexpert.com/a/Magic:number:programming.htm>, [Accessed: April 24, 2008]
- [21] Julie K. Petersen, *The Telecommunications Illustrated Dictionary*, CRC Press, 2002, ISBN: 084931173X.
- [22] Z. P. Zhou, Z. C. Ji, Y. Z. Wang, J. J. Lin, "A New Algorithm of Steganography Based on Palette Image", *1st IEEE Conference on Industrial Electronics and Applications*, May 2006, pp. 1- 4.
- [23] P. Wayner, *Disappearing Cryptography* Second Edition: Information Hiding and Watermarking, *Academic Press, Inc.*, 2002.
- [24] Dinei Florencio and Cormac Herley, "KLASSP: Entering Passwords on a Spyware Infected Machine Using a Shared-Secret Proxy", *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06)*, 2006.
- [25] Roshidi Din and Hanizan Shaker Hussain, "The Capability of Image In Hiding A Secret Message", *Proceedings of the 6th WSEAS International Conference on Signal, Speech and Image Processing*, September 2006.
- [26] L. Y. Por, T. F. Ang and B. Delina, "WhiteSteg: A New Scheme in Information Hiding Using Text Steganography", *WSEAS Transactions on Computers*, May 2008.
- [27] Kefa Rabah, "Steganography – The Art of Hiding Data", *Information Technology Journal* 3 (3), 2004, pp. 245-269.
- [28] Eiji Kwaguchi et. al, "Large Capacity Steganography", *U.S. Patent 6,473,516B1*, Oct. 29, 2002.
- [29] Mohammad Shirali Shareza, "An Improved Method for Steganography on Mobile Phone", *Proceedings of the 9th WSEAS International Conference on Systems*, Greece, 2005, pp. 955-957.
- [30] Hecht, Eugene, *Optics*, 2nd Edition, *Addison Wesley*, 1987.
- [31] Charles G. Boncelet et. al, "Spread Spectrum Image Steganography," *U.S. Patent 6,557,103 B1*, Apr. 29, 2003.
- [32] J. E. Boggess III, P. B. Nation, M. E. Harmon, "Compression of Colour Information In Digitized Images Using an Artificial Neural Network", *Proceedings of the IEEE 1994 National Aerospace and Electronics Conference*, Issue 23-27 May 1994 Page(s):772 - 778 vol.2.
- [33] What is PKI? [Online]. Available: <http://www.entrust.com/pki.htm> [Accessed: August 9, 2008]