MoBiSiS: An Android-based Application for Sending Stego Image through MMS

Rosziati Ibrahim
Department of Software Engineering
Faculty of Computer Science and Information Technology (FCSIT), Universiti Tun Hussein Onn Malaysia (UTHM),
Parit Raja, Johor, Malaysia
rosziati@uthm.edu.my

Law Chia Kee
Department of Software Engineering
Faculty of Computer Science and Information Technology (FCSIT), Universiti Tun Hussein Onn Malaysia (UTHM),
Parit Raja, Johor, Malaysia
qiqilaw1989@msn.com

***Abstract*— A Steganography algorithm is used to hide data from third party in such a way that people are unable to detect the existence of the hidden message inside the stego image. This algorithm is used to maintain the confidentiality of valuable information, and to protect the data from possible sabotage, theft, or unauthorized viewing. Before mobile services, the stego image is sent via e-mail. The recipients have to be connected to Internet and log into their mailbox to download the stego image. This paper introduces a mobile application named MoBiSiS (Mobile Steganography Imaging System). MoBiSiS improves the capability of steganography algorithm by implementing the steganography algorithm for Android-based application. MoBiSiS is able to send the stego image through the Multimedia Messaging Service (MMS) and the stego image can be retrieved from the device's message inbox to extract the hidden message inside the stego image.**

Keywords-steganography algorithm; secret key; image processing.

I. INTRODUCTION

This paper presents an android-based mobile application named MoBiSiS (Mobile Steganography Imaging System). MoBiSiS is a mobile application that is capable of hiding the data inside the image. The image containing the data can then be sent via MMS (Multimedia Messaging Service). MoBiSiS can be used by various users who want to hide data inside an image without revealing the data to other parties. Implementing steganography algorithm [1] in Android-based application makes the usability of steganography increased since mobile is more convenient for user to be brought anywhere and use anywhere. By sending the stego image through MMS, the user is able to get announcement instantly once the stego image received. Therefore, MoBiSiS provides more opportunity for hiding information efficiently.

Steganography is the Greek word for hiding information that invisible to the observer's sense. Steganography is intended to provide secrecy in such a way that others unable to detect the existence of the hidden message. Steganography algorithm helps to hide data and ensures the privacy of the data. This algorithm is used to address digital rights management, conceal secret and protect the confidential information from possible sabotage, theft, or unauthorized viewing.

Steganography algorithm is very important for the purpose of hiding information inside an image. Therefore, the proposed application is being implemented using steganography algorithm to protect the privacy and secrecy of data. The proposed application is an android-based application which allows the user to send or retrieve the hidden data inside the stego image. With a mobile on hand, user can send or retrieve the stego image instantly. The communication media of sending and receiving the steganography image is using the Multimedia Messaging Service (MMS). This application provides an image platform for user to input image, a text box to input the message and allow user to set the key or password of the stego image. Thus, the data is being protected by the key or password.

The rest of the paper is organized as follows. Section 2 reviews the related work and Section 3 presents the details of the implementation of MOBiSiS. Section 4 discusses various results obtained from testing the functionalities of MOBiSiS. The PSNR (Peak signal-to-noise ratio) value of the stego images are also presented and finally, we conclude the paper in Section 5.

II. RELATED WORK

Hiding data is the process of embedding information into digital content without causing perceptual degradation [2]. In data hiding, three famous techniques can be used. They are watermarking, steganography and cryptography. Steganography is defined as covering writing in Greek. It includes any process that deals with data or information within other data. According to Lou *et al.* [3], steganography is hiding the existence of a message by hiding information into various carriers. The major intent is to prevent the detection of hidden information.

Research in steganography technique has been done back in ancient Greek where during that time the ancient Greek practice of tattooing a secret message on the shaved head of a messenger, and letting his hair grow back before sending him through enemy territory where the latency of

 ISBN: 978-1-61208-202-8

this communications system was measured in months [4]. 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. Another method is to mark selected characters within a document by pinholes and to generate a pattern or signature [4]. However, the majority of the development and use of computerized steganography only occurred in year 2000 [5]. The main advantage of steganography algorithm is because of its simple security mechanism. Because the steganographic message is integrated invisibly and covered inside other harmless sources, it is very difficult to detect the message without knowing the existence and the appropriate encoding scheme [6]. There are several steganography techniques used for hiding data such as batch steganography, permutation stehanography, least significant bits (LSB), bit-plane complexity segmentation (BPCS) and chaos based spread spectrum image steganography (CSSIS).

Research in hiding data inside image using steganography technique have been done by many researchers, for example in [1], [7], [8], [9], [10], [11] and [12]. Warkentin *et al.* [7] proposed an approach to hide data inside the audiovisual files. In their steganography algorithm, to hide data, the secret content has to be hidden in a cover message. El-Emam [8], on the other hand, proposed a steganography algorithm to hide a large amount of data with high security. His steganography algorithm is based on hiding a large amount of data (image, audio, text) file inside a colour bitmap (bmp) image. In his research, the image will be filtered and segmented where bits replacement is used on the appropriate pixels. These pixels are selected randomly rather than sequentially. Chen *et al.* [9] modified a method used in [10] using the side match method. They concentrated on hiding the data in the edge portions of the image. Wu *et al.* [11], on the other hand, used pixel-value differencing by partitioning the original image into non-overlapping blocks of two consecutive pixels.

Rosziati Ibrahim et al. [1] propose a steganography algorithm for hiding secret message inside an image. A bitmap (bmp) image is used to hide the data. Data is then embedded inside the image using the pixels. Then the pixels of stego image can then be accessed back in order to retrieve back the hidden data inside the image. Based on the steganogrpahy algorithm in [1], an android-based application is developed to send the stego image through MMS. This android-based application is known as MoBiSis (MoBile Steganography Imaging System). MoBiSiS used the technology of MMS to send or receive the stego images. MMS is a technology that allows a user of a properly enabled mobile phone to create, send, receive and store messages that include text, images, audio and video clips [13]. Users would be able to benefit from the MMS technology for secretly exchange hidden messages and keys, without arousing suspicion of their existence.

III. MOBISIS IMPLEMENTATION

Based on the algorithm proposed in [12], an android-based application is implemented for the purpose of sending the stego image via MMS. This android-based application is written in open source programming language consisting of Java language, Extensible Markup language (XML) and Apache Ant scripting language.

Figure 1 illustrates the activity diagram that represents the flow of activities for proposed application. Activity diagram is one of the Unified Modeling Language (UML) specifications that describe coordination among activities of the application and its external actor by showing the workflow of application. The purpose of activity diagram is to illustrate possible navigation paths through the interface and connections to other parts of the system functionality. However, an activity diagram is differed from a traditional flowchart as it shows concurrency as well as branches of control.

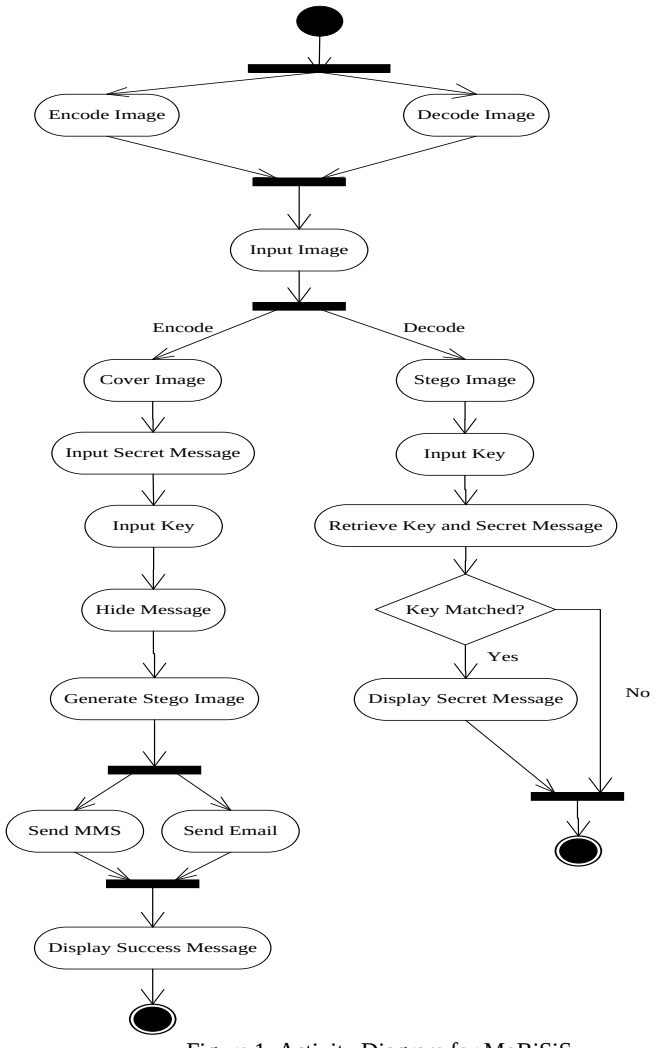

Figure 1. Activity Diagram for MoBiSiS

Based from Figure 1, the application starts with the user selecting to decode message or encode message from menu

indicator, which is an android mobile built in function to perform additional functions. A cover image and a secret message are needed before a key (password) is entered where the key is required to allow generation of stego image. The stego image can be sent through MMS or Email after the stego image is generated. The success message will be displayed and the application ends after the stego image is sent. Stego image and the same key, on the other hand, are needed to retrieve the secret message. Secret message can be retrieved only if the key is matched, or else, the process is failed. Finally, the application ends where the stego image has been generated or secret message has been retrieved and displayed on the text box.

The process of embedding and extracting the message is illustrated in Figure 2.

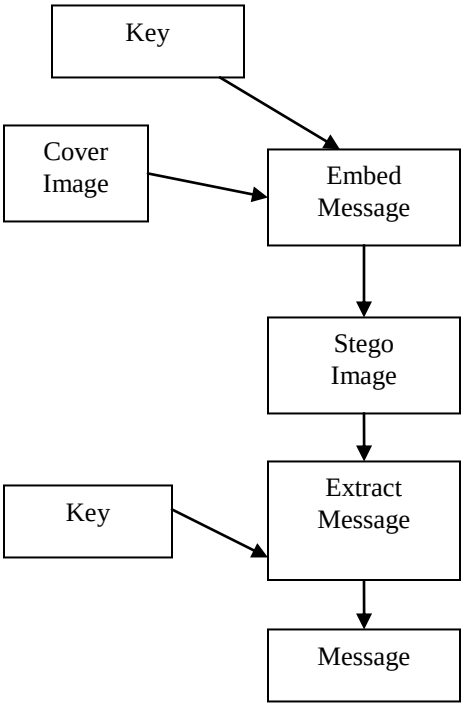

Figure 2. The process of Embedding and Extracting Message

Based from Figure 2, cover image is needed in order to embed the secret message inside the image together with the secret key. Then the message is embedded inside the image. This new image is known as stego image. In order to extract the hidden message inside the stego image, the secret key is needed. Once the correct secret key is provided, the message will be able to retrieve from the stego image. Note that the secret key has to be agreed between the sender and the receiver. If the sender has agreed on a secret key, the sender has to tell the receiver the secret key that has been used for the image. The operational requirements for the application are as follows:

i. Stego image will be generated only after the inputs of cover image, secret message and key (password).
ii. Stego image will be generated in portable network graphic (png) format only.
iii. Secret message will be retrieved only after the input of stego image and the key where the key is matched with the key that has set previously.

MoBiSiS has four functional requirements as stated in Table I. The functional requirements are then used for the functionalities of MOBiSiS.

TABLE I. Functional Requirements of MoBiSiS

i	Generate the Stego Image (Encode)
ii	Retrieve the Secret message from the Stego image (Decode)
iii	Send Steganography Image
iv	Provide additional tools

Based on Table I, the first functional requirement is to generate stego image by the application using the input of cover image, secret message and key from user. The second functional requirement is to retrieve secret message by the application using the input of stego image and the key from user. The third functional requirement is to send the generated stego image through MMS or Email by the application. The forth functional requirement is to enhance the application by providing additional tools for user to interact.

Figure 3 shows the interface of MoBiSiS that allows a user to select the cover image from capture a new photo by mobile camera or from gallery. This page consists of two button which are “From Camera” and “From Gallery” button. The camera function will be switched to on when the “From Camera” button is pressed. On the other hand, the phone’s gallery will be shown when the “From Gallery” button is pressed.

Figure 3. Main Interface of MoBiSiS

Once the image has been choosen (either from camera or from gallery), the secret message can be type and embeded inside the selected image. Figure 4 shows the interface for this process.

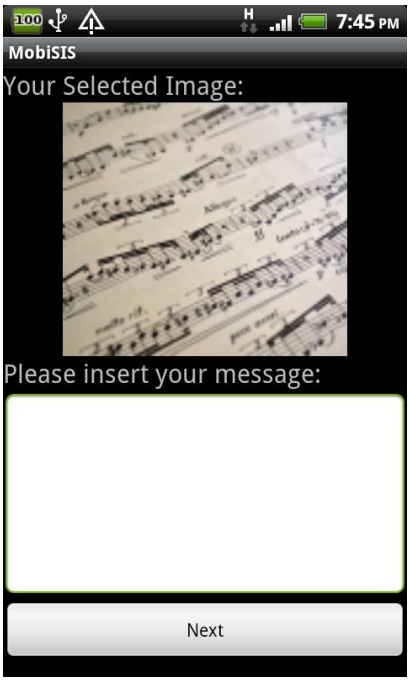

Figure 4. Interface to type the message for embedding

The process flow for MoBiSiS is shown in Figure 1. Once the information has been stored inside the stego image, this stego image can be sent via MMS or Email without exposing the information embeds in the stego image. The hackers would not be able to retrieve information inside the image. The information can only be retrieved from the stego image with the system (MoBiSiS) installed in the mobile and the secret key for the image. Figure 5 shows the option of sending the stego image via MMS or Email.

IV. RESULTS AND DISCUSSION

The functionalities of MoBiSiS are then tested using various images. Figure 6 shows some of the cover images that have been used for testing. These images are used to embed the secret message, send the stego images via MMS and retrieve the secret message**.** Note that the images are used to test the steganography algorithm used in MoBiSiS.

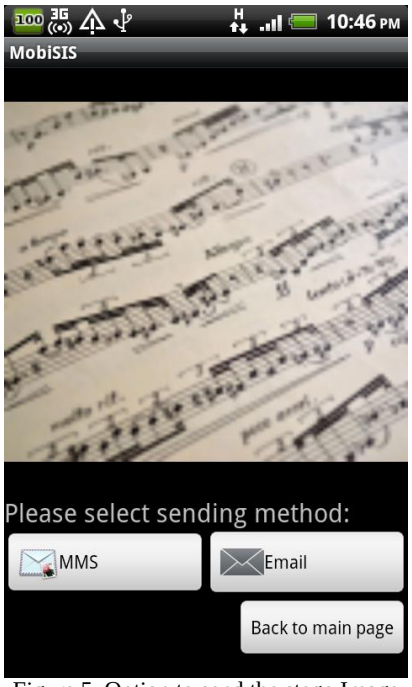

Figure 5. Option to send the stego Image

Kidnap Person

Police Officer

Dog

Rabbit

Figure 6. Images that are used for Hiding Data

We also tested the stego image for its PSNR (Peak signal-to-noise ratio). PSNR is a standard measurement used in steganograpy technique in order to test the quality of the

stego images. The higher the value of PSNR, the more quality the stego image will have.

If the cover image is *C* of size *M x M* and the stego image is *S* of size *N x N*, then each cover image *C* and stego image *S* will have pixel value *(x, y)* from *0* to *M-1* and *0* to *N-1* respectively. The PSNR is then calculated as follows:

$$PSNR = 10.\log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (1)$$

where

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(C(x,y) - S(x,y))^2$$

Note that MAX is the maximum possible pixel value of the images. For example, if the pixels are represented using 8 bits per sample, then the MAX value is 255.

If the stego image has a higher PSNR value, then the stego image has better quality image. Table II shows the PSNR value for stego images in Figure 6. The PSNR is calculated using the equation of PSNR in (1).

TABLE II. The PSNR Value of Stego Images

Image	PSNR for Stego Image
Kidnap Person	70.7586
Police Officer	69.0479
Dog	74.6493
Rabbit	72.6493

Based on Table II, the PSNR value shows that the stego images have higher value, which confirms that the quality of the stego image is still high.

The image file format used for MoBiSiS can be in JPEG, GIF, PNG and BMP format which supported by mobile application. However, since MMS is only compatible in JPEG image file format, the generated stego images are in JPEG format. Note that, MMS only support to send image size that less than 30 Kilo Bytes (KB) to maintain its actual size and pixel. Hence, the proposed application compresses the image which larger than 10 KB in order to generate a stego image which will not exceed 30 KB. However, Huffman Encoder assigns shorter codes for characters that appear more often and longer codes for characters that appear less often. Thus, the shorter code improves the capacity of hiding character inside the stego image. To increase as much as characters that can be hidden, zip technique is used to reduce to total size of file and to enhance the security of the file. Table 3 shows the comparison of different image file format and different image size by using MoBiSiS. These JPEG, GIF, PNG and BMP images are used as cover images to encode the zipped file within it.

V. CONCLUSIONS

This paper discusses an android-based application named MoBiSiS (Mobile Steganography Imaging System). MoBiSiS has been developed using the steganography algorithm proposed in [1]. The algorithm used has been tested in term of the quality of the stego image and its PSNR value. The application of steganographic algorithm has been enhanced to mobile application. MoBiSiS can be used by users who want to hide the data inside the image without revealing the data to other parties. MoBiSiS maintains privacy, confidentiality and accuracy of the data.

ACKNOWLEDGMENT

This research is supported under the Fundamental Research Grant Scheme (FRGS) Vot 0738.

REFERENCES

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

[2] Chen M., Memon N., and Wong E.K. (2008). Data Hiding in Document Images. In Nemati H. (Ed.). *Premier Reference Source – Information Security and Ethics: Concepts, Methodologies, Tools and Applications*, Volume 1, Chapter 1.32. New York: Information Science Reference. pp. 438-450.

[3] Lou D.C., Liu J.L., and Tso H.K. (2008). Evolution of Information – Hiding Technology. . In Nemati H. (Ed.). *Premier Reference Source – Information Security and Ethics: Concepts, Methodologies, Tools and Applications*, Volume 1, Chapter 1.32. New York: Information Science Reference. pp. 438-450.

[4] Schneider (2000). *Secrets & Lies*, Indiana:Wiley Publishing.

[5] Cole E. (2003). *Hiding in Plain Sight: Steganography and the Art of Covert Communication.* Indianapolis: Wiley Publishing.

[6] Jahnke T. and Seitz J. (2008). An Introduction in Digital Watermarking Applications, Principles and Problems. In Nemati H (Ed). *Premier Reference Source – Information Security and Ethics: Concepts, Methodologies, Tools and Applications*, Volume 1, Chapter 1.42, New York: Information Science Reference. pp. 554-569.

[7] Warkentin M., Schmidt M.B., and Bekkering E. (2008). *Steganography and Steganalysis.* Premier reference Source – Intellectual Property Protection for Multimedia Information technology, Chapter XIX, pp. 374-380.

[8] El-Emam N.N. (2007). *Hiding a Large Amount of Data with High Security using Steganography Algorithm.* Journal of Computer Science 3 (4), pp. 223-232.

[9] Chen P.Y. and Wu W.E. (2009). *A Modifed Side Match Scheme for Image Steganography.* International Journal of Applied Science & Engineering 2009, 7, 1:53-60.

 ISBN: 978-1-61208-202-8

[10] Chang C.C. and Tseng H.W. (2004). *A Steganographic Method for Digital Image using Side Match.* Pattern Recognition Letters 25 2004, pp. 1431-1437.

[11] Wu P.C. and Tsai W.H. (2003). *A Steganographic Method for Images by Pixel-Value Differencing.* Pattern Recognition Letters 24 (2003), pp. 1613-1626.

[12] Rosziati Ibrahim and Teoh Suk Kuan, (2010). *Steganography Imaging System (SIS): Hiding Secret Message inside an Image,* Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2010, WCECS 2010, 20-22 October, 2010, San Francisco, USA, pp. 144-148.

[13] Jain, Y. K., Kumar, R., and Agarwal, P. (2011). Securing data using jpeg image over mobile phone. *Global Journal of Computer Science and Technology*, 11(13), pp. 5-6.

TABLE III. Comparison of different image file format in different image size

IMAGE FILE FORMAT	**FILE SIZE**			**HIDE MESSAGE**	**RETRIEVE MESSAGE**	**RETRIEVE MESSAGE AFTER MMS**	**DISTORTION**
	COVER IMAGE	ZIPPED FILE	STEGO IMAGE				
JPEG	3.06 KB	179 bytes	3 KB	√	√	√	No
JPEG	3.06 KB	223 bytes	2 KB	√	√	√	No
JPEG	3.06 KB	236 bytes	Failed	—	—	—	—
JPEG	9.08 KB	179 bytes	16 KB	√	√	√	No
JPEG	9.08 KB	223 bytes	16 KB	√	√	√	No
JPEG	9.08 KB	236 bytes	Failed	—	—	—	—
GIF	3.08 KB	179 bytes	2 KB	√	√	√	No
GIF	3.08 KB	223 bytes	2 KB	√	√	√	No
GIF	3.08 KB	236 bytes	Failed	—	—	—	—
GIF	9.05 KB	179 bytes	10 KB	√	√	√	No
GIF	9.05 KB	223 bytes	10 KB	√	√	√	No
GIF	9.05 KB	236 bytes	Failed	—	—	—	—
PNG	3.11 KB	179 bytes	Failed	—	—	—	—
PNG	9.08 KB	179 bytes	3 KB	√	√	√	No
PNG	9.08 KB	223 bytes	3 KB	√	√	√	No
PNG	9.08 KB	236 bytes	Failed	—	—	—	—
BMP	3.05 KB	179 bytes	1 KB	√	√	√	No

 ISBN: 978-1-61208-202-8