

Implementation of Steganography Image System on Android Using LSB Embedding Method

Yodiaz Gilby Dhilega^{1,3}, Danial Md Nor^{1,2*}, Istiadi ST. MT.³

¹Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 84600, MALAYSIA

²Multi Network Systems Focus Group,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 84600, MALAYSIA

³Teknik Elektro,
Universitas Widyagama, Malang, INDONESIA

DOI: <https://doi.org/10.30880/eeee.2020.01.01.039>

Received 15 July 2020; Accepted 06 September 2020; Available online 30 October 2020

Abstract: Steganography implementation in digital security is still used today as an effort to maintain data security. Least Significant Bit is implemented on the android device as an important aspect of digital security application and for easier use of the method. Least Significant Bit is dependent on the size of the image, pixel composition, and file size to maintain the data embedded into each last bit of the pixel. The idea of this research is to build the implementation of Least Significant Bit using Steganography Image System on the Android Platform. Thus, in this, every research aspect of the stego image is tested to check whether the image could maintain its embedded data through various ways of direct cybersecurity attack while measuring the quality of stego image with the cover image by using Java Language, which extremely simple and powerful high-level programming language. From six samples with different resolutions, vary from 120px width to 184px, the stego image was tested by inserting 16 string characters into the cover image. The Least Significant Bit is able to embed the string characters into cover image successfully without making any changes that perceivable by human eyes. The PSNR method is used to verify the digital image properties from the prepared sample. The result shows that the average value of PSNR of 120px to 184px width stego image is 20 dB.

Keywords: Steganography, Image Processing, Least Significant Bit

1. Introduction

Since time immemorial, humans have always been involved in secrecy in matters that are personal to them. Over time, humans began to recognize various techniques and algorithms to perfect confidential communication, one of them is Steganography[1]. Steganography is the science and art of hiding secret messages in such a way that no one can find or suspect the existence of such messages. Unlike cryptography, where cryptography functions to hide the contents of the message so that it cannot

*Corresponding author: daniel@uthm.edu.my
2020 UTHM Publisher. All rights reserved.
penerbit.uthm.edu.my/proceeding/index.php/eeee

be read by third parties or opponents. Besides being used in the covert exchange of operation, the usage of steganography on the other grounds such as copyright, prevent e-document forging[2].

In its implementation in the modern world, steganography is still used today. However, the implementation is in a different media, namely electronic media. Concealment of messages with electronic media can be in various ways, can be in the form of text, audio, video, even pictures. Message security in this era is very much needed because, with the development of the internet, interception and retrieval of information unilaterally are also very common. To prevent information leakage and maintain message integrity, a system is needed to maintain the integrity of messages and prevent information leakage so that privacy can be maintained and private communication cannot be known by others or third parties.

It is clear that from the problem posed before, it is necessary to create an easy steganography system to maintain the security and credentials of messages to stay safe based on Android so that their 'practical' use in everyday life becomes easier. So, by using Algorithm Steganography Image System and LSB (Least Significant Bit), it is expected that data security can be maintained and data leakage can be prevented to provide convenience and comfort for those who communicate.

The objective of this study is to study the required parameters and variable for Android Steganography System, design and build an appropriate android-based steganography system using the Least Significant Bit method to get the optimal solution in preventing confidential data leaks, and analyze the Stego Image of Least Significant Bit using Peak Signal-to-Noise Ratio for effectiveness for each of the image tested.

This study is focused on developing a model, plan, and apps for Steganography Image System using the Least Significant Bit method for more secure data transactions using Android OS. Variables such as images, image type extensions, resolution, media, types of android that are supported, to the quality of steganographic data that will be modeled by Android Studio which will be used as an IDE to integrate various factors into one part. At the end of this project, the output of the application will be analyzed to determine the quality of the resulting image after going through the LSB data encryption process and comparing the difference with media that previously used images with different resolutions

2. Materials and Methods

The methodology approach and implementation for developing Steganography Image System is using the Least Significant Bit method. The main objectives is to develop the Steganography Image System using the Least Significant Bit method for hiding the properties of text inside and image media.

The primary implementation of the study are in the form of image and the testing is via Matlab 2016a. All materials will be described based on their functionality in research of this project.

2.1 PNG (portable network graphics)

PNG (Portable Network Graphics) format is used widely in the world. It is capable of compression and transparency support, not like JPEG. Since PNG compression is completely lossless and since it supports up to 48-bit truecolor or 16-bit grayscale. Hence, saving, restoring, and then the resaving image will not impair its quality

2.2 Matlab

Matlab is an interactive application software for numeric computation and data visualization. This software is used to determine the value of matrix before and after embedding of stegotext on three image file formats by calculating the value of entropy, mean, and light intensity of each type to see the differences of information. This software provides the validation of structural information for a large portion of the samples produced for this research.

Lists using items marked with a,b,c, or i, ii, iii, and others can also be considered. Items in the list should be indented similar to paragraph indentation.

2.3 Implementation of steganography using least significant bit method

The process of embedding implements the Least Significant Bit method to insert the message into a digital image and the process is divided into two parts which are encrypt and decrypt. The insertion of LSB is the modification of Teoh Suk Kuan & Rosziati Ibrahim[3]. The Encryption process for LSB insertion will be implemented in the following algorithm. First, the message for media insertion that prepared before will be compressed, because data type string that is compressed will be harder to detect and to be read significantly rather than plain insertion. Text usually chosen because it will not look suspicious as it were inside media[4].

And after string compression, it will be encrypted with a password from the prepared variable. Meanwhile, for the final process, an encrypted message will be encoded into the image. The embedding proses will use Least Significant Bit to encode the data into a prepared image. After the encoding is complete, the process will stop.

For the Decryption algorithm, the embedded message from media will be decoded from encrypted media using LSB decoding. After that, the compressed string text will be decoded using the password. After the text is decoded, the message will be decompressed to obtain the original message from embedded media.

2.3 MSE and PSNR calculation

The quality of image medium after the steganography process would not have too much difference compared to before. After the steganography process, the image quality will not have a drastic change, because of that, the observer will not realize that there is a secret message inside the media. To measure the steganographic imagery require an objective test to each media used as a sample. The measurement is done by calculating PSNR and MSE values objectifically.

PSNR (Peak Signal to Noise Ration) is a ratio between the maximum value of the measured signal and the amount of noise that affects the signal. In this research, PSNR is used to compare the quality before and after the image was processed. The PSNR values are measured in decibels(dB).

To measure the PSNR, the value of MSE (Mean Square Error) must first determined because Mean Square Error is the average square error of value between the processed image and cover image. The following formula is for Mean Square Error ::

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)] \quad \text{Eq. 1}$$

$I(x,y)$ = Value of pixel on cover image M = Height of stego image (in pixel)

$I'(x,y)$ = Value of pixel on stego image N = Width of stego image (in pixel).

After the value of MSE obtained, then the value of PSNR could be calculated from the square of maximum value divided by MSE[12]. The following formula is for PSNR :

$$PSNR = 10 . LOG \left(\frac{MAXi^2}{MSE} \right) \quad \text{Eq. 2}$$

MSE = The value of MSE $MAXi$ = The maximum value of the image used

3. Results and Discussion

The results of the Steganography Image System using Least Significant Bit method will be processed to determine the result of analysis. The stego image sample which will be processed using LSB (Least Significant Bit) method is prepared with the dimension of each sample is different for each sample based on the previous study of Gies Masita Arini et. Al[5]. The media image that prepared before will be transformed from steganography applications and later will result in the output of processed stego images.

Using the Least Significant Bit, the color will no have a significant change because L only changes the value of the byte one higher or one lower from the previous value. Because of that, the significant change of color does not change much, because human eyes cannot distinguish small changes by the LSB Method.

There are two sections of the Steganografi application, which is Encrypt and Decrypt. To embed text media into a cover image, the layout for Encrypt.java provides all required variables and parameters to perform the LSB method. However, the layout only limits text to be inserted into the cover image

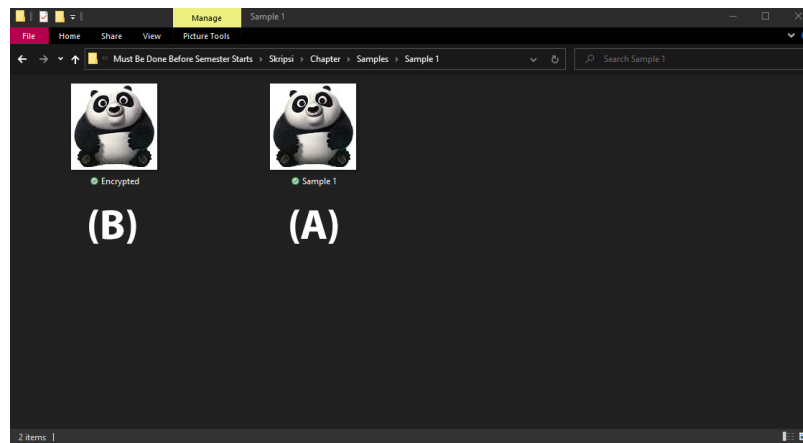


Figure 1 : Samples of cover image (A) and stego Image(B).

Figure 1 shows the after and the result after the embedding process of steganography. The test was run by inserting 16 character string into the cover media. The result doesn't show much significant change in color except the digital image properties which will be analyzed later after the process proceeds. The table for prepared sample with stego image comparison can be seen on table 1.

Table 1: Comparison of cover Image with processed stego image



File Name	File Dimension	MD5 Checksum	SHA1 Checksum
Sample 1	120x120	6899D7D3E2DF1A006A0B270 C4650D99C	F1AF7A514D566AC204168F40 89B07C6FDB30BA97
Encrypted	120x120	0E147484D144293420B9512F6 DEB11A4	5E987896A117EC6F1E27F60E BDACBC9A16DBF5FD

3.1 Properties analysis

The properties analysis of LSB was done by using the Matlab 2016a program. Matlab is used to detect and calculate the properties of both encrypted and original samples to declare the value of properties using the PSNR and SNR method. The original sample is first analyzed to check its properties value of PSNR and the value of MSE to compare the result with the processed image.

Table 2 shows the result of PSNR, MSE value using a function in Matlab with added noise for image comparison.

Table 2: Properties of original image & stego image value

File Image	File Name	File Dimension	PSNR	SNR
	Sample 1	120x120	20.9171	17.8247
	Encrypted	120x120	20.6810	17.5881

The analysis on all samples is to check the PSNR value of each sample and stego image for comparison. The result of each sample will be analyzed using Matlab and the result is shown on table 3:

Table 3: Properties of original image & stego image value of each sample

File Name	PSNR	SNR
Sample 2	20.5764	19.1372
Encrypted	20.2555	18.8157
Sample 3	20.9991	18.7932
Encrypted	21.3156	19.1090
Sample 4	20.1971	19.6091
Encrypted	20.2870	19.6988
Sample 5	21.0184	19.2636
Encrypted	20.8602	19.1083
Sample 6	21.1193	17.7692
Encrypted	20.9066	17.5558

From table 3 above, it can be seen that the average value of PSNR from all samples listed above is 20dB when the sample was tested by inserting 16 string characters into the cover image. This concludes that embedding with inserting a string into cover media does not produce a value that is too different.

3.2 Image integrity test analysis

There is three image integrity test that was tested on the system, which is image rotation, image resizes, and brightness & contrast increase to the image in PNG format. For the test, it will be run on Android 8.1 Oreo and the image will be rotated, resize, and brightness contrast increase using Adobe Photoshop CC 2015 for the analysis.:

The image will be rotated 180° for analysis. The sample is from sample 1 to sample 6, with all sample format is PNG and it is already encrypted with string character inside it. The result is shown in table 4.

Table 4: Image rotation testing

File Name	Number of Testing	Result
Encrypted 1	5 Times	Message Corrupted
Encrypted 2	5 Times	Message Corrupted
Encrypted 3	5 Times	Message Corrupted
Encrypted 4	5 Times	Message Corrupted
Encrypted 5	5 Times	Message Corrupted
Encrypted 6	5 Times	Message Corrupted

For the next testing, The image will be resized into 50px width for analysis. The sample is from sample 1 to sample 6, with all sample format is PNG and it is already encrypted with string character inside it. The image was resized using Windows 10 Microsoft Paint. The result is shown in table 5.

Table 5: Image resize testing

File Name	Number of Testing	Result
Encrypted 1	5 Times	Message Corrupted
Encrypted 2	5 Times	Message Corrupted
Encrypted 3	5 Times	Message Corrupted
Encrypted 4	5 Times	Message Corrupted
Encrypted 5	5 Times	Message Corrupted
Encrypted 6	5 Times	Message Corrupted

The image brightness/contrast will be increased into 25 value of Photoshop Brightness/Contrast setting for analysis. The sample is from sample 1 to sample 6, with all sample format is PNG and it is already encrypted with string character inside it. The image brightness was increased using Adobe Photoshop CC 2015. The result is shown in table 6

Table 6: Image brightness/contrast testing

File Name	Number of Testing	Result
Encrypted 1	5 Times	Message Corrupted
Encrypted 2	5 Times	Message Corrupted
Encrypted 3	5 Times	Message Corrupted
Encrypted 4	5 Times	Message Corrupted
Encrypted 5	5 Times	Message Corrupted
Encrypted 6	5 Times	Message Corrupted

From all three tests that have been conducted on the system, the decryption of rotated image, resized image, or increased contrast image were failures. The system is not able to keep the message compilation intact if there is an attack on the image. When the image is corrupted, the decryption function is not able to work properly due to the condition of the image. Therefore, after the image undergoes the process of transformation, the information on the image becomes lost or incomplete.

4. Conclusion

At the end of this research, the implementation of the Steganography Image System using the Least Significant Bit method is investigated using various observation and implementation. The analysis is done to determine is the PNG file format can be used for steganography standard. Tools used in this

research are Matlab 2016a Academic License, Microsoft Windows 10 Paint, Android Studio, Android Virtual Device, and Adobe Photoshop CC 2015.

Different sample with each resolution of a sample is different, Ranging from 120px width to 184px height. Based on the obtained results, the following finding could be deduced, The color of the stego image compared with the original image is not perceivable by human eyes since the Least Significant Bit and when the cover image is transformed into a stego image, the stego image is different from the cover image.

This can be proven by comparing MD5 or SHA1 of cover image with a stego image, Stego image that was tested with 16 string characters will produce a 20dB average of PSNR. The test was done by examining 6 different samples with minimal 5 number of testing for each sample, and The Least Significant Bit Stego Image cannot be decrypted if the original value of the image is changed in various ways, for example, increasing the Contrast/Brightness, Resizing the image, or rotating the image.

Acknowledgement

The authors would also like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] I. Jonathan, A. Y. Haryono, and K. Leonardi, "Penelitian Mengenai Metode Steganografi Least Significant Bit," *J. Ultim. Comput.*, vol. 9, no. 1, pp. 17–20, 2017, doi: 10.31937/sk.v9i1.569
- [2] S. Channalli and A. Jadhav, "Steganography An Art of Hiding Data," *Int. J. Comput. Sci. Eng.*, vol. 1, no. 3, pp. 137–141, 2009
- [3] R. Ibrahim and T. S. Kuan, "Steganography Algorithm to Hide Secret Message inside an Image," *Comput. Technol. Appl.*, vol. 2, pp. 102–108, 2011
- [4] N. Laila and A. S. R. Sinaga, "Implementasi Steganografi LSB Dengan Enkripsi Vigenere Cipher Pada Citra," *Sci. Comput. Sci. Informatics J.*, vol. 1, no. 2, p. 47, 2019, doi: 10.22487/j26204118.2018.v1.i2.11221
- [5] G. M. Arini and T. I. Widyawan, "Pengamanan Pesan Steganografi dengan Metode LSB Berlapis Enkripsi dalam PHP," *Pengamanan Pesan Steganografi dengan Metod. LSB Berlapis Enkripsi dalam PHP*, vol. 3, pp. 11, 2012