Real Time Implementation of Secured Multimedia Messaging Service System using Android

Geetanjali R. Kshirsagar[*], Savita Kulkarni[**]

[*] Department of Electronics, Maharashtra Institute of Technology, Pune.
[**] Department of Electronics and Telecommunication, Maharashtra Institute of Technology, Pune.

Abstract- In this paper encryption and steganography algorithms are implemented using JAVA™ with android platform to provide the security for real time multimedia messaging service system. Establishing hidden communication for mobile has become an important subject of security. One of the methods to provide security is steganography. Steganography is used to hide secret information inside some carrier. To improve the security, encrypted secret data will be hidden inside MMS. The image is made to be hidden into image from MMS which provides more secured transmission than text information embedded into the MMS. The Least Significant Bit (LSB) embedding technique is used to hide the secret information (image). Different sizes of secret images are taken and later the calculations have been done for the PSNR (Peak Signal to Noise Ratio) of image in MATLAB. Encryption and steganography algorithms are ported on HTC Desire mobile device with android version 2.2.3.

Index Terms- Android Platform, Encryption, LSB, MMS, PSNR, Security, Steganography.

I. Introduction

After rapid growth of the Internet, establishing hidden communication is an important subject of security that has gained increasing importance. Telecommunication companies started to add additional features to their mobile phones such as MMS (Multimedia Messaging Service) in order to attract more customers. And users can securely communicate its secrets by means of sending and receiving MMS messages. One of the most popular uses of mobile phones has been the exchange of messages between users. The Short Messaging System (SMS) was introduced with GSM mobile phones and it very rapidly became popular among users. The Multimedia Messaging System (MMS) offers the ability to send and receive multimedia content using a mobile phone. Now a day, most of the mobile phones not only are capable of sending and receiving Multimedia Messages (MM), but also contain an embedded camera and can run customized applications (e.g. using Java 2 Platform Micro Edition, J2ME).

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. One of the main and relatively new hidden communication methods is steganography. In steganography the data are hidden in a cover media so that other persons will not notice that such data is there. This is a major distinction of steganography method with the other methods of hidden exchange of information such as cryptography and can be mainly applied to media such as images, text, video clips, music and sounds. The combination of both may give the best results, as a message can be encrypted before it is hidden into another object. Steganography concerns itself with ways of embedding a secret message into a cover object, without altering the properties of the cover object evidently.

The encryption, transforming message (here used image) into cipher text (encoded form) and decryption, a reverse process, plays an important role in concealing the confidentiality of the message. The message is first encrypted with a key during an encryption process and then hiding it in available format (here used image). Thus sending an encrypted message increases the security level of the message. Once received the message need to be decrypt using same key [1] which implies the concept of symmetric key steganography in which the key is symmetric for both sender and receiver. The paper introduce an approach which enhance the security of data by first encoding it , hiding in cover medium and sent it to the intended recipient.

II. PROBLEM DEFINITION

The aim of the project is to hide the data as an image over an image from MMS using least significant steganographic algorithm and before hide an image perform encryption on it. Send the stego file to the destination where the retrieving of the secret image is done on mobile device with Android.

Problem Solution

The proposed method should provide better security while transferring the data or messages from one end to the other end. The main objective of the project is to hide encrypted secret image into an image from MMS which acts as a carrier file having secret data and to transmit to the destination securely without any modification. If any distortions occur in the image or on its resolution while inserting the secret message into the image, there may be a chance for an unauthorized person to modify the data. So, the data (image) encryption at sender and decryption at receiver and steganography plays an important role in this project.

Proposed System Architecture

The data hiding patterns using the steganographic technique in this project can be explained using this simple block diagram. The block diagram for steganographic technique is as follows.

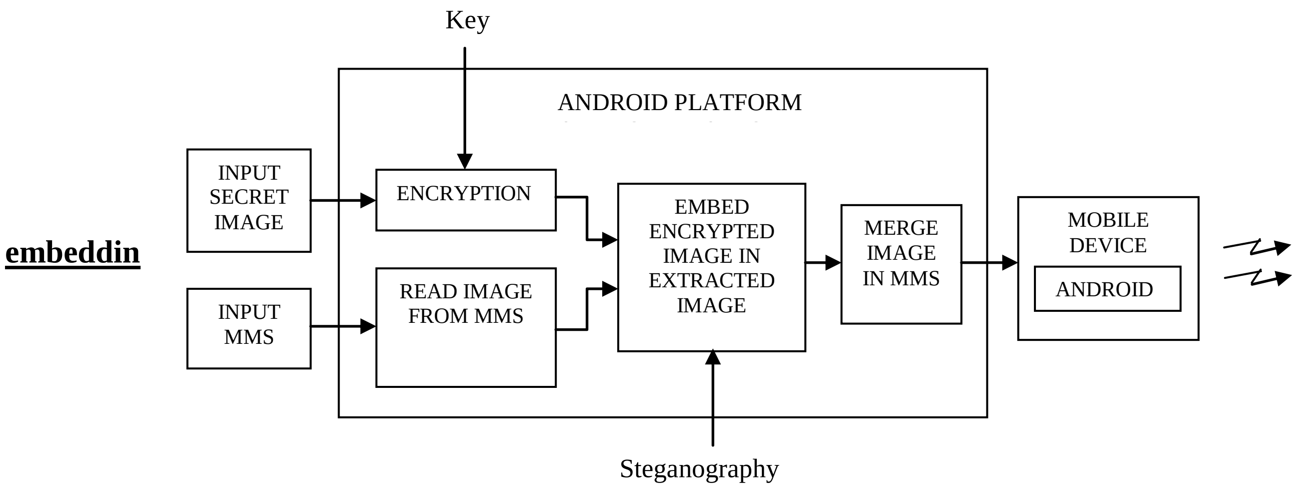

Figure 1. Block Diagram of Embedding Process in Binary Image

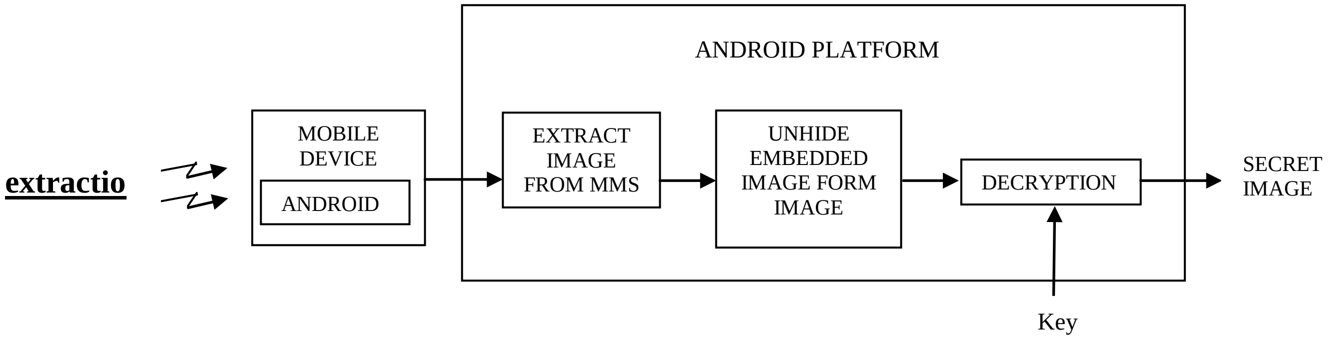

Figure 2. Block Diagram of Extracting Process from Binary Image

III. IMAGE STEGANOGRAPHY ALGORITHM

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colours of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover. For JPEG, the direct substitution of stenographic techniques is not possible since it will use lossy compression. So it uses LSB substitution for embedding the data into images.

Each of these pixels in an image is made up of a string of bits. We can commandeer the 4-least significant bit of 24-bit true color image to hold 4-bit of our secret message (image) by simply overwriting the data that was already there. By experimental, we note that: The impact of changing the 4-least significant bits will be minimal and indiscernible to the human eye.

4-LSB Embedding Algorithm

1. Start
2. Read Multimedia Message from MicroSD card of an Android mobile device.
3. Extract an image from Multimedia Message

Cover image = Extracted image

4. Convert the cover image into stream of binary bit.
5. Make 4 LSB of each byte zero.
6. Read a Secret image from Android mobile MicroSD card.
7. Encrypt Secret image using key, PRNG and encryption algorithm with Android platform.
8. Convert Encrypted Secret image into stream of binary bit.
9. Read the lower nibble of Encrypted Secret image byte.
10. Hide lower nibble of Encrypted Secret image byte into the lower nibble of blue channel pixel byte of Cover image.
11. Read the upper nibble of Encrypted Secret image byte.
12. Hide upper nibble of Encrypted Secret image byte into next blue channel byte of the Cover image.
13. Go to step 9 till to hide complete encrypted secret image
14. Merge an image with hidden encrypted image (stego image) into MMS.
15. Send MMS.

As given in algorithm secret image byte is divided into two nibble. Hide each nibble into the blue channel byte of cover image.

The minimum size of Cover image = 10* Size of Secret image + n

We add n pixels because we don't set secret data in the header of cover image; we start setting secret data after the header of cover image. We just presented the used algorithm to hide an image file in a JPEG image. In the next section, we present the extracting algorithm.

4-LSB Extracting Algorithm

Extracting the secret image data is performed by reversing the process used to insert the secret message in the cover image. The following steps describe the details of extraction process.

1. Read Multimedia message.
2. Extract the image from Multimedia message i.e. stego image
3. Read lower nibble from first pixel of blue channel from stego image which is the lower nibble of secret image.
4. Read lower nibble from second pixel of blue channel from stego image which is the upper nibble of the secret image.
5. Now this is the byte of secret image.
6. Repeat the step 3 and step 4 to read all byte of secret image.
7. Decrypt secret image using same key.
8. Display retrieved secret image.

For measuring the quality of reconstructed image as compared to the original image, the metric needs to be define. There are three common error metrics used for estimating noise on images are RMSE, PSNR, and SSIM. The PSNR is defined as:

PSNR = 10 x log10 x (MAX / RMSE)

Where, MAX is the maximum pixel value of the image. In the case of 8 bits gray scale images the MAX value will be 255.

IV. IMPLEMENTATION

The cover images and secret images which want to send/transfer are stored in the MicroSD card of Android mobile device with android version 2.2.3. Cover image along with text message is Multimedia Message. Access both the images (cover and secret) from sdcard using android platform and follow the following steps on android platform to perform steganography.

A. The first part which is an encryption /decryption process is implemented in JAVA™ with android platform and the output of encryption is save in a file which is needed at the time of hiding the content in an image.
B. In the next phase, the encrypted file and cover image is taken as input file. The file is hiding in the image and this implementation is done in JAVA™ with android platform.
C. At the receiver side follows exactly opposite steps. First unhide an image and then decrypt it.
D. The last phase is the PSNR calculation and analysis which is implemented in MATLAB [11]. The function code is written in Editor Window for PSNR calculation which is executed and the PSNR for the two images – original secret image at sender and retrieved secret image at receiver.

V. RESULT

We implemented the proposed algorithms to hide a secret image into cover image. Results of developed algorithms for encryption and steganography are shown in Table I and Table II.

EXAMPLE 1:

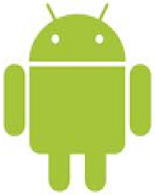

Figure 4.Android Logo

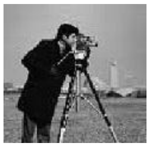

Figure 5. Cameraman Secret image

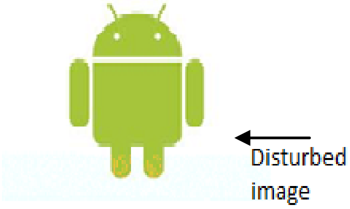

Figure 6: Stego Image

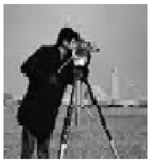

Figure 7: Retrieved Secret Image

Table I: PSNR calculation of various size secret images For Example 1(Android Logo)

Cover Image Size	Secret Image Size	PSNR
171 X 131	100 X 100	60.4065
171 X 131	120 X 120	57.7510
171 X 131	125 X 125	55.6609

EXAMPLE 2:

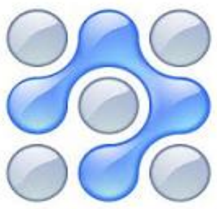

Figure 8: Bubbles Cover image Figure 9: Secret image

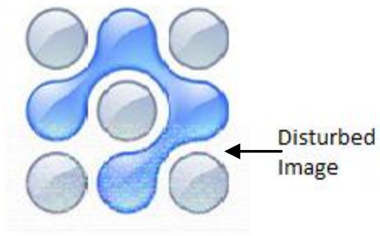

Figure10:Stego image Figure11:Retrieved secret image

Table II: PSNR calculation of various size secret images For Example 2 (Bubbles)

Cover Image Size	Secret Image Size	PSNR
150 X 150	112 X 112	65.2210
150 X 150	125 X 125	62.7042
150 X 150	140 X 140	59.5214

VI. CONCLUSION

4-LSB substitution is successfully implemented to hide secret image into an image from MMS which provides the security during transmission of MMS. Algorithm is developed on android platform and tested by porting same on actual android mobile device HTC Desire with android version 2.2.3. It gives PSNR results as varying size of secret image. From the results it is observed that noise in stego image increases as size of secret image increases and PSNR should be greater than 55 to transfer image successfully. In this way encryption and steganography security algorithms are successfully implemented using Android platform with high potential of security.

ACKNOWLEDGMENT

I would sincerely like to thank my guide **Prof. Mrs. Savita Kulkarni** for her guidance, timely help and valuable suggestions without which this seminar report would not have been possible. I would sincerely like to thank **Prof. (Dr.) B. C. Gargash** for his guidance, timely help for completion of this seminar. I am very much thankful to **Prof. (Dr.) G. N. Mulay** for their guidance along with the inspiration and proper suggestions without which this seminar would not have been completed. I would also like to thank Electronics and Telecommunication department and the staff for their co-operation and help in completion of the seminar report. Also, I thank all my friends for their appraisal and criticism, which has helped me to make my seminar a success.

REFERENCES

[1] 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

[2] Rosziati Ibrahim, Law Chia Kee “MoBiSiS: An Android-based Application for Sending Stego Image through MMS” ICCGI 2012 : The Seventh International Multi-Conference on Computing in the Global Information Technology June 24-29, 2012 - Venice, Italy

[3] S.Mohanapriya “Design and Implementation of Steganography Along with Secured Message Services in Mobile Phones” International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 5, May 2012

[4] Mrs. Kavitha, Kavita Kadam, Ashwini Koshti, Priya Dunghav ”Steganography Using Least Signicant Bit Algorithm” Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 3, May-Jun 2012, pp. 338-341

[5] V. Lokeswara Reddy, Dr. A. Subramanyam, Dr.P. Chenna Reddy ” Implementation of LSB Steganography and its Evaluation for Various File Formats” Int. J. Advanced Networking and Applications Volume: 02, Issue: 05, Pages: 868-872 (2011)

[6] B.V.Rama Devi, P.Prapoorna Roja, D.Lalitha Bhaskari and P.S.Avadhani , “A New Encryption Method for Secure Transmission of Images” International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010, 2801-2804

[7] B.N.Jagadale, R.K.Bedi,Sharmishtha Desai,” Securing MMS with High Performance Elliptic Curve Cryptography” International Journal of Computer Applications (0975 – 8887)Volume 8– No.7, October 2010

[8] http://www.developers.android.com

[9] http://www.intechopen.com

[10] http://www.mathworks.com

AUTHORS

First Author – B.E. (Electronics and Telecommunication), M.E. (Electronics - Digital System) Appeared. Maharashtra Institute of Technology, Pune., Email address: kshirsagar.geetanjali@gmail.com
Second Author – B.E. (Electronics), M.E. (Computer Engineering), Associate Professor Maharashtra Institute of Technology, Pune. Email address: savita.kulkarni@mitpune.edu.in