ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 6.047

Volume 5, Issue 4, April 2017

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Image Steganography using Encrypted Message

Dr. Asoke Nath[1]
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata – India

Sayudh Roy[2]
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata – India

Chahat Gopalika[3]
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata – India

Debayan Mitra[4]
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata – India

Abstract: ***Steganography is the art of hiding a message so that a would-be eavesdropper is unaware of the message's presence. The goal is to hide the existence of the message from unauthorized personnel. It presents a task of transferring the embedded information to the destination without being detected by the attacker. Many applications have already been developed for Steganography. In the present work, the authors propose to enhance the application's portability and ease of use, and thus have developed the application on the Android platform. Nath et al. have already proposed various ways to encrypt and hide secret messages in different cover files. In this study, the secret message was encrypted using an algorithm inspired from the Vigenère cipher technique and then the message was embedded into the LSBs of the bytes of the cover file. It means to hide one byte of a secret message; the authors used the LSBs of eight bytes of the cover file without destroying the property of the file significantly. The proposed bit exchange is reversible, i.e. the decryption is done in the reverse way of the encryption. The authors applied the present steganography algorithm on image files and the result found was satisfactory.***

Keywords: Image Steganography, Android, LSB Substitution, Polyalphabetic Substitution, Encryption, Decryption.

I. INTRODUCTION

Steganography is a special method of writing hidden messages in such a way that no one apart from the sender and the receiver can even realize that there is a hidden message. For example, the sender may embed a big text file in an image in such a way that there should not be any significant change in the image. The extent of steganography can be stretched to even embedding some voice or image or text in any host file which may be of any media file type.

The aim of this work is to implement **Image Steganography** on any format of image files for Android devices. The levels of security provided to the users is increased by encrypting the messages to be embedded, using a polyalphabetic substitution cipher technique which has been derived from the Vigenère Cipher method. Our objective is to come up with a technique of hiding the message in the image file in such a way, that there would be no perceivable changes in the image after the message insertion. At the same time, if the message that is to be hidden were encrypted, the level of security would be raised to quite a satisfactory level.

To encode and embed the information in the Android Application, the authors have used the following procedure:

i) Encrypt the entered message using the polyalphabetic substitution algorithm developed, with the key entered by the user.

ii) Embed the secret message using the LSB Substitution Algorithm in the image and generate corresponding stego image for the same.

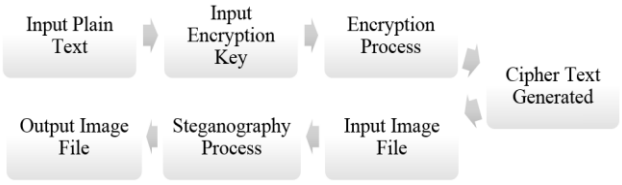

Figure 1: Encryption and Embedding Process

To extract and decode the information in the Android Application, the authors have used the following procedure:

i) Extract the secret message from the stego image using the Reverse Algorithm as used during embedding.

ii) Verify authenticity of the key entered for decryption and decode the secret message using the key. Although, an incorrect key will yield an output, but it will not be the desired message.

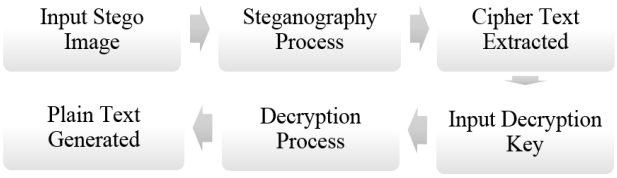

Figure 2: Extraction and Decryption Process

In LSB steganography, the least significant bits of the cover media's digital data are used to conceal the message. The simplest of the LSB steganography techniques is LSB replacement. LSB replacement steganography flips the last bit of each of the data values to reflect the message that needs to be hidden.

Consider an 8-bitgrayscale bitmap image where each pixel is stored as a byte representing a gray scale value. Suppose the first eight pixels of the original image have the following gray scale values:

11010010	01001010	10010111	10001100
Byte 1	***Byte 2***	***Byte 3***	***Byte 4***
00010101	01010111	00100110	01000011
Byte 5	***Byte 6***	***Byte 7***	***Byte 8***

To hide the letter C whose binary value is 10000011, we would replace the LSBs of these pixels to have the following new grayscale values:

Table 1: LSB Substitution

Before	Bit	After	Remarks
11010010	1	1101001**1**	Change
01001010	0	0100101**0**	No Change
10010111	0	1001011**0**	Change
10001100	0	1000110**0**	No Change
00010101	0	0001010**0**	Change
01010111	0	0101011**0**	Change
00100110	1	0010011**1**	Change
01000011	1	0100001**1**	No Change

II. PROPOSED ALGORITHMS

A. Message Encryption Algorithm:

Step 1: Start

Step 2: Input the plain text message and the key for encryption

Step 3: The plain text is converted into the cipher text using the extended form of the Vigenère cipher algorithm using all 256 characters

Step 4: The encryption can be described using the following formula: $C_i = T_i + K_i \pmod m$; where C_i: i-th character of the cipher text, T_i: i-th character of the open text, K_i: i-th character of the key phrase and m: length of the alphabet (which in this case is 256)

Step 5: If the key phrase is shorter than the open text, which is usual, than the key phrase is repeated to match the length of the open text

Step 6: End

B. Message Decryption Algorithm:

Step 1: Start

Step 2: Input the cipher text message and the key for decryption

Step 3: The cipher text is converted into the plain text much alike the previous algorithm by only reversing the calculations

Step 4: The decryption can be described using the following formula: $T_i = C_i - K_i \pmod m$; where the abbreviations are same as the previous algorithm

Step 5: If the key phrase is shorter than the cipher text, which is usual, then the key phrase is repeated to match the length of the cipher text

Step 6: End

C. Data Embedding Algorithm:

Step 1: Extract the bytes of the cover image

Step 2: Extract the characters of the encrypted message

Step 3: Skip a few bytes of the cover image to maintain its integrity

Step 4: Replace the least significant bit of each of the bits of the image bytes with each of the least significant bits of the message bytes

Step 5: Repeat step 4 till all the message characters has been embedded

Step 6: Place some terminating symbol to indicate end of data

Step 7: Obtained stego image

Step 8: End

D. Data Extraction Algorithm:

Step 1: Extract the bytes of the stego image

Step 2: Build characters by finding the least significant bits of each of the image bytes and storing them in arrays of size 8

Step 3: Once the array is full, it converts to the character representation using the ASCII standard and then clears itself for the next set of bits to be extracted

Step 4: Follow steps 2 and 3 till up to terminating symbol, otherwise go to step 5

Step 5: The secret message has been extracted

Step 6: End

III. Results and Discussions

Sl. No.	Cover File			Secret Message
	Name	Image Format	Size	
1.	input1.bmp	BMP	769 KB	This is our Project.
2.	input2.png	PNG	26 KB	This is PNG testing
3.	input3.jpg	JPEG	156 KB	This is final testing

Case (1): Bitmap (BMP) File Input

Cover File + *Secret Message*: This is our Project. = *Output File*

769 KB

199 KB

Case (2): Portable Network Graphics (PNG) File Input

Cover File + *Secret Message*: This is PNG testing = *Output File*

26 KB

16.6 KB

Case (3): Joint Photographic Experts Group (JPEG) File Input

Cover File + *Secret Message*: This is final testing = *Output File*

156 KB

101 KB

The authors applied the present steganography algorithm on image files as illustrated above and the results found were satisfactory. To embed any message inside a cover file the user has to enter a password to encrypt the message and the same password is to be used to decrypt the secret message.

IV. Conclusion and Future Scope

Steganography is an effective way to hide sensitive information and the relevance of such information hiding techniques is perpetually on the rise in this modern digitized world where the threat of data theft and interception are part of everyday reality.

In this paper the authors seek to address this constant threat of data theft by making the information that is sent across digital networks more secure and seek to do that with the help of image steganography. There are different methods to implement the process of encryption via image steganography. The Least Significant Bit Substitution method used here, takes the least significant bit of the image and replace it with the least significant bit of the encrypted message, and this process is continued progressively to obtain the stego image.

Keeping in mind the easy availability and widespread use of the Android platform, the steganography application has been developed on the same which can be used to encrypt and decrypt messages via image steganography and then used for private communication.

The authors have observed some slight changes in picture resolution in the case of comparatively sizeable embedded messages and this is a shortcoming intended to be looked into and rectified in future works. Furthermore, to maintain the integrity of the image, a few initial bytes of the image had to be skipped to ensure that the image's header was not tampered with. However this leads to a slight exception where one cannot encrypt a message in an image whose size is less than or equal to the number of bytes skipped in the program code, otherwise the encrypted message will not be embedded as it will run out of bytes. So preferably, it is helpful to use images of considerable size. The authors intend to continue working on these errors and iron out these glitches in the future.

ACKNOWLEDGEMENT

The authors are grateful to all the professors of the Department of Computer Science in St. Xavier's College for helping the authors to finish this work.

References

1. Agniswar Dutta, Abhirup Kumar Sen, Sankar Das , Shalabh Agarwal and Asoke Nath: New Data Hiding Algorithm in MATLAB using Encrypted secretmessage : Proceedings of IEEE CSNT-2011 held at SMVDU (Jammu), 03-06 Jun, 2011
2. Android: https://developer.android.com/reference/packages.html
3. Palak R Patel and Yask Patel: Survey on Different Methods of Image Steganography: IJIRCCE Vol. 2, Issue 12, December 2014
4. Advanced Steganography Algorithm using encrypted secret message, Joyshree Nath and Asoke Nath, International Journal of Advanced Computer Science and Application (IJACSA) Vol-2 No.3, Page 19-24, March (2011)
5. An Overview of Image Steganography by T. Morkel, J. H. P. Eloff and M. S. Oliver
6. An Overview of Steganography by Shawn D. Dickman

AUTHOR(S) PROFILE

Dr. Asoke Nath, is an Associate Professor in the Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. Dr. Nath is involved in research work in Cryptography and Network Security, Steganography, Green Computing, Mathematical modelling of social networks, Big Data Analytics, Cognitive Radio, Data Science, e-learning, MOOCs etc. Dr. Nath has already published more than 211 publications in different journals.

Sayudh Roy, is a final year B. Sc. Computer Science (Hons.) student from St. Xavier's College (Autonomous), Kolkata. He has an avid interest in the field of Cryptography and is looking to pursue his Masters in the areas of Machine Learning and Artificial Intelligence.

Chahat Gopalika, is a final year B. Sc. Computer Science (Hons.) student from St. Xavier's College (Autonomous), Kolkata. Apart from her keen interest in Computers and Designing, she is looking to pursue Civil Services and aspires to be in the Indian Administrative Service.

Debayan Mitra, is a final year B. Sc. Computer Science (Hons.) student from St. Xavier's College (Autonomous), Kolkata. Apart from his ardent passion in Cryptography and Steganography, he is interested in public speaking and debating and is looking to a future with Business Administration in Marketing.

