An Android Application to find the Sender Information of Fraud Image using Steganography

Anjan H K[1], Anudeep K R[1], Revanth Kumar C A[1] and Dr. Purohit Shrinivasacharya[2]

[1]Students, Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumakuru
Email-id: anjan01april@gmail.com, krandeep50@gmail.com, revanthkumar803@gmail.com

[2]Associate Professor, Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumakuru
Email-id: purohitsn@gmail.com

Abstract - Now days, the social media usage has been increasing and user are using images and videos instated of text message. In this images or videos some are relevant and some are fraud images/videos. Spreading these types of images/videos creates problems like data pack stealing, personal information capturing, false information about health etc. In this project we are trying to solve these problems by using steganography to identify the senders. We are extracting some useful information from the mobile/PC and adding it into an image/video without knowing the sender. This information will help to identifying the sender by decoding the steganography image. Finally, we conduct some experiment and tabulate the results to compare the original image and corresponding steganography image. The existing application can adopt this technique for identifying the sender.

Keywords - Steganography, Stealing, False Information, Image, Sender

I. INTRODUCTION

Steganography is a creation of media which contains secret information. Nowadays digital steganography usage has been increased to send coded information from one to another. The information is embedded into digital texts or multimedia (images, videos, music, etc.) to create steganography media. The digital steganography and steganoalysis (an art of unveiling the hidden information) are the branch of modern science and are evolving new ideas. The new ideas or old ideas are used to develop the applications. In this paper, we are using the existing techniques to embedded the secret information to identify the fraud image distributor. Steganography and steganalysis have been used to develop an android application for finding the sender of the fraud image.

A. General Stenographic System

Steganography has its set of specification like other science. The original message which does not contain the secret message is called as the cover message. The secret message is embedded into the cover message, then this new message is called as stego message. The embedding of the secret message into a cover message is done by using some sorts of algorithms or techniques. The cover may be a single or multiple files. The algorithm puts the secret information into the cover directly, instead of putting the data directly we can encrypt the secret message and store into cover. Figure 1 shows the overview of generic Steganography system.

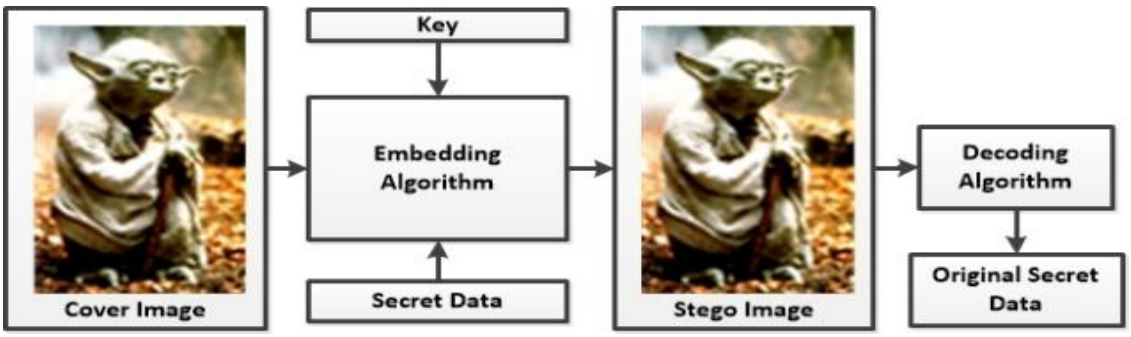

Fig. 1 An overview of a generic Steganography system

There are many forms of input will accept the embedding algorithms, but it requires some rules that have to meet the embedding process. They are as follows:

- The perversion of the cover message as a result of the embedding algorithm should be as invisible as possible.
- The part of the secret message should not contain in the header of the stego file. The embedded message must become part of the cover message and should be immune to manipulation attacks such as resampling or filtering.

- It is good practice to put error checking codes into the stego message to extract the embedded message and correct if any damage.
- It is necessary that the original cover message should not reach to an eavesdropper or be used twice. Because the embedding process is additive and embedded message can be recovered if an eavesdropper has different stego files which employ the same cover message.

II. PROPOSED SYSTEM

The proposed system is to embed the secret information extracted from the mobile/social media is shown in Figure 2. The architecture consists of three models are as follows:

- Extraction of identity form sender
- LSB Embedding Algorithm to hide identity data
- LSB Decoding Algorithm to extract the identity data from an embedded media.

The identity information is extracted from a mobile or social media as user name, IP address, MAC address, date, time and email ID. This information is considered as secret information to embed into an image.

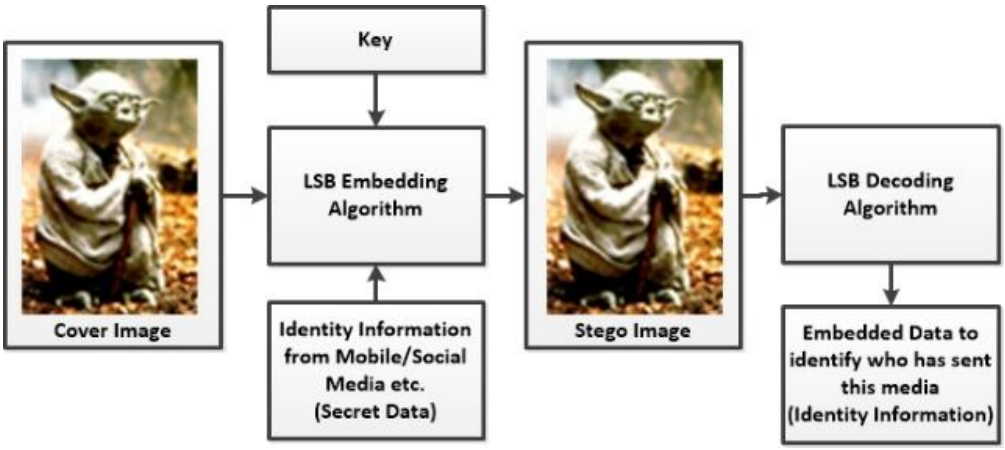

Fig. 2 Architecture of proposed system

A. Least Significant Bit (LSB) Modification

List significant bit modification is the traditional technique to embed the secret message into a cover message. In this method, every byte of a least significant bit is replaced with secret message one-bit information. In this proposed method, secret information is embedded into an image. Normally images are in RGB color model, so each pixel of an image consists three bytes. These three bytes are used to embed the secret information bits. This technique is most effective against human detection because the value will change as zero or one. The color mixing component also not gives more difference once it generates as the image.

Algorithm:

Step 1: The given cover image is converted into an RGB components like 24 bits of 3 bytes.

C(i, j) = Image(p, q)

Where i=0 to M-1, j= 0 to (N-1)*3 and p= 0 to M-1, q=0 to N-1

Step 2: Convert the identity information extracted from device into bits representation

SM = Bits (Identity information)

Step 3: Assign k=0 repeat the process until all bits of SM is embedded

Step 4:

$$S(i,j) = \begin{cases} \mathrm{C(i,j)} - 1, \text{if } \mathrm{LSB(C(i,j))} = 1 \text{ and } \mathrm{SM}_k = 0 \\ \mathrm{C(i,j)} + 1, \text{if } \mathrm{LSB(C(i,j))} = 0 \text{ and } \mathrm{SM}_k = 1 \\ \mathrm{C(i,j)}, \text{if } \mathrm{LSB(C(i,j))} = \mathrm{SM}_k \end{cases}$$

where S is the stego image

Step 5: Repeat Step 3 and 4 process until all bits of SM is embedded

Step 6: Convert the S values into combined RGB component and store as an image.

Example: In this proposed system, text information is embedded into every color channel of an image. The image is converted into a three channels like Red (R), Green (G) and Blue (B) and replace every byte least significant bit by text information bit. The Table 1 shows the example of each channel and embedding message bits replace indicated with bold black color.

Message First byte: (Assume it is '9') – 00111001

TABLE 1. LSB MODIFICATION WITH EACH CHANNEL VALUES AND EMBEDDING MESSAGE

Pixel #	Channel	Channel Value	Replaced Channel Value
1	R	11111000	1111100**0**
	G	11001001	1100100**0**
	B	00000011	0000001**1**
2	R	11111000	1111100**1**
	G	11001001	1100100**1**
	B	00000011	0000001**0**
3	R	11111000	1111100**0**
	G	11001001	1100100**1**
	B	00000011	00000011

B. Flow diagram

The proposed system architecture flow diagram is shown in Fig. 3.

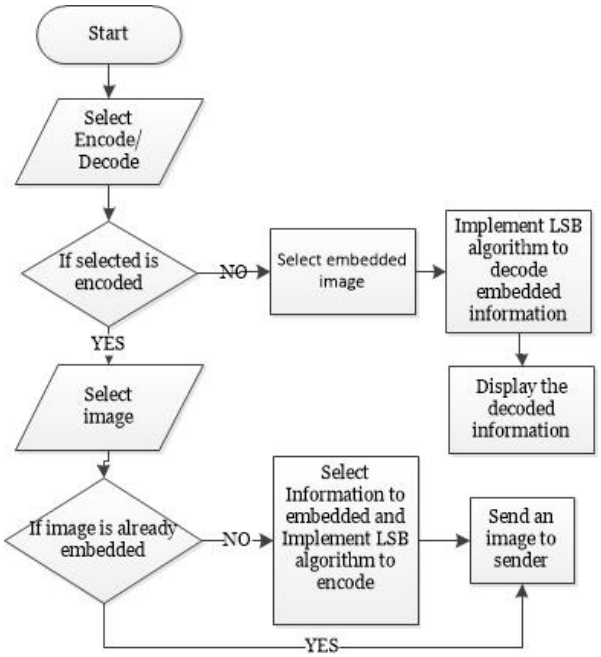

Fig. 3 Flow diagram of proposed system

III. EXPERIMENTAL RESULTS

The proposed system has been tested by submitting the different images in different mobiles. The Table 2 shows the image number and corresponding image size before and after embedding the personal information of the sender. Table 2 also contains the time to perform the embedding process.

TABLE 2. SAMPLE IMAGES AND ITS CORRESPONDING SIZES BEFORE AND AFTER EMBEDDING AND TIME

Image	Before encoding size (in KB)	After encoding size (in KB)	Time required (in seconds)
Image 1	154	155	4
Image 2	230	232	5
Image 3	135	135	3
Image 4	160	163	6
Image 5	146	147	3
Image 6	170	172	4
Image 7	222	224	4
Image 8	284	287	5
Image 9	188	190	3
Image 10	120	121	5

Figure 4 shows the main screen of the designed system and it contains two buttons namely decode and encode. If user selected the encode button it does encode activity before sending and this activity is shown in Figure 5. User is not aware of the encoding with his/her information. This process can be applied to any applications to identify the sender when something went wrong. Here, we have demonstrating by developing separate application by adding encode and decode buttons. To decode the information contained in the image, we have provided the button to decode the particular information available in an image as shown in Figures 6 and 7.

Fig. 4 Main Activity in these application

Fig. 5 After embedding the information into an image

Fig. 6 Embedded image and extracted embedded information

IV. Conclusion

In this paper, we have developed an application to find the fraud image sender information using steganography. The LSB algorithm is used to encode the sender details before sending image. The process of encoding takes less time and it will be not known to the user. The information we are encoding in an image are name, IP address, MAC address, e-mail id, date and time. Finally, we have tested for different image and compared the stored information against different mobiles. This application helps to control the crime and to identify the crime for cyber analysis. There is no much difference between cover image and stego image, so sender will not get any doubt. Hence these application provides efficient way of finding the sender of the fraud image.

References

[1] Soni Ashish and Jain, Jitendra and Roshan, Rakesh, "Image steganography using discrete fractional Fourier transform," in *Proc. of Intelligent Systems and Signal Processing (ISSP)*, pp. 97-100, 2013.

[2] Parmar Ajit Kumar Maganbhai , Prof. Krishna Chouhan, "A Study and literature Review on Image Steganography", *International Journal of Computer Science and Information Technologies*, vol. 6, no. 1, pp. 685-688, 2015.

[3] Achsah Elizabeth Varghese, Reconfigurable Processor for Image Steganography using DCT with Morphological Operations, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 9, pp. 8053-8061, 2015.

[4] Ravi S P and Dhanalakshmi L, M., "DWT and Modified AES based Secure Image Steganography on ARM A8 Processor," *International Journal of Engineering Research & Technology*, vol. 4, no. 5, pp. 1482- 1486, 2015.

[5] Vivek Kumar Yadav and Sonu Lal, "A Survey on Video Steganography based on Information Concealment using Abstraction and Rework Domain," *International Journal for Scientific Research & Development*, vol. 4, no. 9, pp. 976-981, 2016.