IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 3, Issue 2 (Sep. – Oct. 2013), PP 40-46
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Secret Image Sharing Using S-box

Guda Uma Shanker[1], Tejas Tirupati[2], Sri Harsha Thudi[3]
[1], [2], [3] *(ECE, Sree Nidhi institute of science and technology, JNTU-H, India)*

Abstract: *Stenography is the art of hiding secret information in other information and transmitting via wires or through free space. Many variants to transfer the secret image already exist currently but further advancements of digital technology has made the transfer much more effective and faster without tampering the original data. In this paper, we propose a method that encodes a secret image in Grey scale imageinto another set of values and then merged into the container image which is an RGB image and transmitted to the desired location. The image containing the secret image hardly differs from the original container image, which makes decoding the original image difficult.*

Keywords:*Image Hiding, Secret Sharing, Stenogaphy*

I. Introduction

With the growing usage of internet in every field of application, there is a need for protecting the secret information that is to be transmitted over the internet. The secret information can be transmitted in secured way by converting the data format which is called encryption. The main drawback of this technique is that the data is not hidden; there are chances of decrypting the data encrypted into the original form by a person with decent knowledge of cryptography. Stenography is one of the possible solutions to above mentioned disadvantage by providing the secrecy to the information.

Secret text and images often exist in the military and commercial applications. Transmitting secret images over the electronics lines and through the computer networks always carries a danger of eavesdropping and increases the chances of tampering the data which could lead to exposure of secret information. The images of higher resolution have got more number of colours and more information which need to be transmitted long distances. The direct transmission of such images over networks increases the chances of tampering the data. Thus, there is a need for manipulating the secret image to safeguard it against the eavesdropping, tampering and to increase the safety. To solve this flummox situation, image hiding proves to be one of the possible solution for transferring the images or secret text.

II. Related Works

The most suitable area for stenography is image on which many methods have been proposed and designed. The main reason is there are the chances of hiding the secret information into another image or source without drawing the attention of the human visual system. In this respect, a number of techniques have been developed using features like

1. Substitution
2. Masking and Filtering
3. Transform Techniques

The main feature of substation technique is that it does not increase the size of the file. Depending on the content of the hidden image it can eventually cause a tinge from the unmodified version of the image. The masking and the filtering techniques use the stats of the image. The Transform technique has been employed in embedding the message by the modulating coefficients in a transform domain.

III. Proposed Design

In this section we propose a method for image hiding where we store one image file called secret image in another image called carrier image. The primary objective of our technique is to provide more security and concurrently using less storage. Digital images are of two types i) 8 bit images and ii) 24 bit images. The main intention is not to change the visual properties of both images after reconstruction and carrier image after embedding the secret image. In our design we propose a technique to hide the 8 bit image into the 24 bit image. The image hiding can be described in the following steps

3.1 Encryption

3.1.1 Thresholding
3.1.2 Encoding

3.1.3 Scrambling
3.1.4 Embedding

3.2 Decryption
3.2.1 De-embedding
3.2.2 De-scrambling
3.2.3 Decoding

The above steps are explained in detail below:

3.1 Encryption

3.1.1 Thresholding:

The carrier image pixels need to be manipulated before it is further used for embedding the secret image; the Thresholding values are dependent on the encoding technique usage. According to this proposed design the image lower pixel value is set at 8 and the maximum value is set at 247 .i.e. the pixel values above 247 are set at 247 and for pixel values which are less than 8 are set at 8.

3.1.2 Encoding:

The grey value of the particular pixel is converted into the 8-bit binary representation, thus the grey value is converted into 8 bits, which is further divided into two parts of 4 bits each. The pixel value in the grey scale is converted into the 8 bit and the every four bit is encoded using the following look up table.
The following table is used for encoding

Table1. Look up table

Binary Number	Equivalent Number
1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1111	8

3.1.3 Scrambling

To increase the security and minimizing detecting the secret image, we randomize the pixel instead of directly embedding the image into the carrier image after encoding it. Many scrambling techniques are known, but both the ends of the communication must decide which one to use. The chaos process is done in such a way that the whole image is divided into two blocks and the horizontal pixels are scrambled the above process is repeated for the divided two blocks; it is carried till the block size becomes 2. After the horizontal pixels are scrambled the same procedure is carried out for the vertical columns.

3.1.4 Embedding

Every pixel of the secret image is spit into two halves and encoded based on above the lookup table and added to the carrier image's pixel value the second half of the secret image pixel can be added to any other plane of the RGB plane or any other location in the same plane or in the any other plane and the rest of the pixel values follows that above sequence. Thus, the final image is ready.

3.2 Decryption

3.2.1 .De-embedding

It is done by subtracting the carrier image pixel value from the final image pixel value. If the carrier image pixel value is higher than the final image pixel value, the obtained value needs to be complemented to get the original value and a flag is added. If the final image value is greater than the carrier image value, the obtained value is true in nature and no flag is added.

3.2.2 De-scrambling

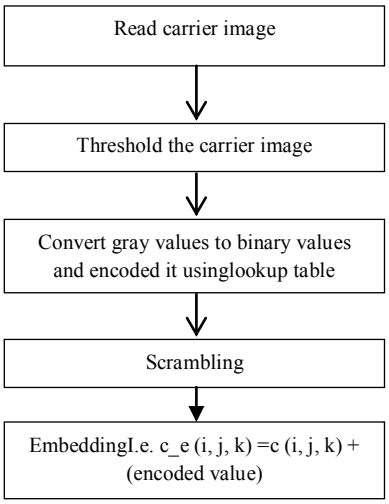
Depending on the scrambling technique agreed by the both the ends of communicating parties, we apply the same scrambling technique in reverse way to decode the secret image.

3.2.3 Decoding

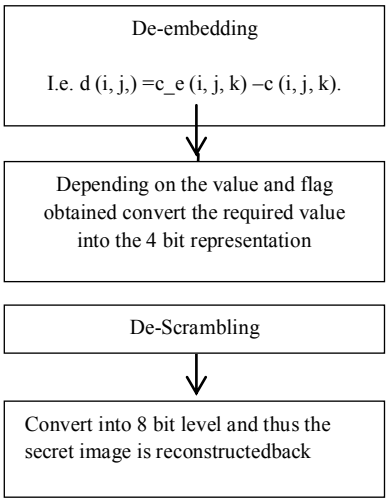
Depending on the values obtained from the de-embedding stage, each encoded value is replaced with the equivalent binary representation obtained from the above look up table, and the values with the flag are bit complemented. The first half is added back to the second half pixel value with the necessary multiplications i.e. by converting the binary number to the decimal values.

Flow chart:

Encoding:



Decoding:



The above process is illustrated below, using the pixel values which are a part of my test image.
Let the initial values of the carrier images (RGB image) can be seen from table 2

Table 2 Initial values of Carrier Image

Plane	Initial values	Threshold values
1.(R)	$\begin{bmatrix} 123 & 213 & 245 & 250 & 26 \\ 34 & 55 & 0 & 2 & 8 \\ 254 & 213 & 25 & 4 & 9 \\ 85 & 96 & 250 & 54 & 7 \\ 25 & 45 & 8 & 9 & 246 \end{bmatrix}$	$\begin{bmatrix} 123 & 213 & 245 & 245 & 26 \\ 34 & 55 & 8 & 8 & 8 \\ 245 & 213 & 25 & 8 & 9 \\ 85 & 96 & 245 & 54 & 8 \\ 25 & 45 & 8 & 9 & 245 \end{bmatrix}$
2.(G)	$\begin{bmatrix} 12 & 21 & 24 & 252 & 25 \\ 32 & 50 & 06 & 96 & 45 \\ 32 & 245 & 8 & 89 & 90 \\ 85 & 96 & 25 & 58 & 70 \\ 25 & 40 & 80 & 99 & 255 \end{bmatrix}$	$\begin{bmatrix} 12 & 21 & 24 & 245 & 25 \\ 32 & 50 & 08 & 96 & 45 \\ 32 & 245 & 8 & 89 & 90 \\ 85 & 96 & 25 & 58 & 70 \\ 25 & 40 & 80 & 99 & 245 \end{bmatrix}$
3.(B)	$\begin{bmatrix} 8 & 20 & 25 & 25 & 25 \\ 30 & 50 & 60 & 96 & 44 \\ 30 & 250 & 20 & 30 & 89 \\ 86 & 9 & 27 & 56 & 75 \\ 28 & 124 & 158 & 158 & 250 \end{bmatrix}$	$\begin{bmatrix} 8 & 20 & 25 & 25 & 25 \\ 30 & 50 & 60 & 96 & 44 \\ 30 & 245 & 20 & 30 & 89 \\ 86 & 9 & 27 & 56 & 75 \\ 28 & 124 & 158 & 158 & 245 \end{bmatrix}$

Let the secret image pixel values are as follows, along with its binary representation.

Table3. Encoding process for an image as are follows.

Initial values	$\begin{bmatrix} 121 & 120 & 5 & 2 & 8 \\ 25 & 63 & 25 & 45 & 89 \\ 12 & 2 & 3 & 45 & 2 \\ 8 & 9 & 4 & 89 & 12 \\ 48 & 255 & 89 & 45 & 6 \end{bmatrix}$
Binary representation	$\begin{bmatrix} 01111001 & 01111000 & 00000101 & 00000010 & 00001000 \\ 00011001 & 00111111 & 00011001 & 00101101 & 01011001 \\ 00001100 & 00000010 & 00000011 & 00101101 & 00000010 \\ 00001000 & 00001001 & 00000100 & 01011001 & 00001100 \\ 00110000 & 11111111 & 01011001 & 00101101 & 00000110 \end{bmatrix}$
Encoded values	$\begin{bmatrix} 7 & -6 & 7 & -7 & 0 & 5 & 0 & 2 & 0 & -7 \\ 1 & -6 & 3 & 8 & 1 & -6 & 2 & -6 & 5 & -6 \\ 0 & -3 & 0 & 2 & 0 & 3 & 2 & -2 & 0 & 1 \\ 0 & -7 & 0 & -6 & 0 & 4 & 5 & -6 & 0 & -3 \\ 3 & 0 & 8 & 8 & 5 & -6 & 2 & -6 & 0 & 6 \end{bmatrix}$

Column scrambling	$\begin{bmatrix} 0 & -7 & 2 & 0 & 5 & -7 & 0 & -6 & 7 & -6 \\ 5 & -6 & -6 & -6 & 2 & 8 & 1 & 3 & 1 & -3 \\ 0 & 1 & -2 & 3 & 2 & 2 & 0 & 0 & 0 & -3 \\ 0 & -3 & -6 & 4 & 5 & -6 & 0 & 0 & 0 & -7 \\ 0 & 6 & -6 & -6 & 2 & 8 & 5 & 8 & 3 & 0 \end{bmatrix}$
Final scrambled values	$\begin{bmatrix} 5 & -6 & -6 & -6 & 2 & 8 & 1 & 3 & 1 & -3 \\ 0 & -3 & -6 & 4 & 5 & -6 & 0 & 0 & 0 & -7 \\ 0 & 6 & -6 & -6 & 2 & 8 & 5 & 8 & 3 & 0 \\ 0 & 1 & -2 & 3 & 2 & 2 & 0 & 0 & 0 & -3 \\ 0 & -7 & 2 & 0 & 5 & -7 & 0 & -6 & 7 & -6 \end{bmatrix}$
After embedding to plane 1(R)	$\begin{bmatrix} 130 & 207 & 252 & 238 & 6 \\ 35 & 49 & 11 & 16 & 9 \\ 245 & 210 & 25 & 10 & 9 \\ 85 & 89 & 245 & 48 & 8 \\ 28 & 45 & 16 & 15 & 250 \end{bmatrix}$
Embedded values to plane 2.(G)	$\begin{bmatrix} 17 & 21 & 26 & 245 & 18 \\ 28 & 52 & 2 & 101 & 39 \\ 35 & 247 & 6 & 89 & 91 \\ 89 & 101 & 19 & 58 & 67 \\ 19 & 42 & 74 & 99 & 251 \end{bmatrix}$

IV. Experimental Results

The above process is implemented and the following results were obtained. The following is the secret image which needs to be transmitted

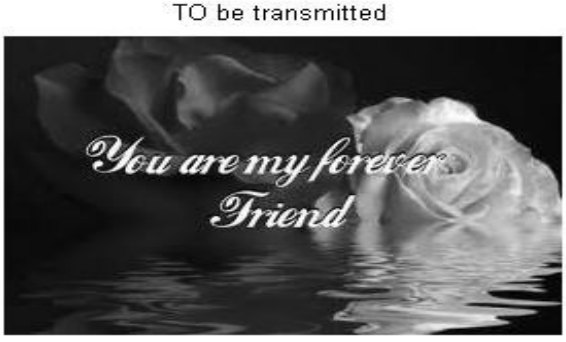


Fig 1. Secret Image to be transmitted

The following are different carrier images

Fig 2. Carrier Image (1)

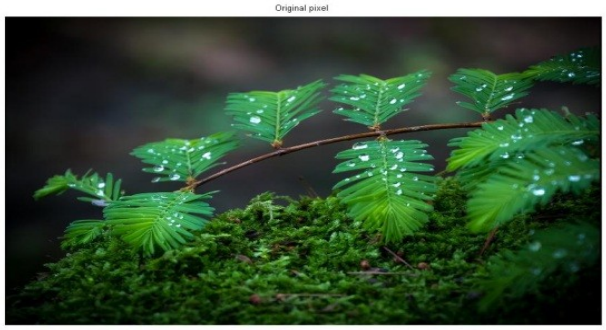


Fig 3. Carrier Image(2)

The images obtained after embedding the images are as follows

Fig 4. Embedded Image(2)

Fig 5. Embedded Image(1)

The reconstructed image is as follows:

Fig 6. De-embedded Image

V. Conclusions

We proposed a method such that secret image can be shared by embedding into another image and then transmitted; the benefit of this technique is that the secret image cannot be decoded until we know the scrambling technique involved and the embedded image picture is very much like the original image, thereby making the secret image transfer easier without getting notified.

References

[1] Secret image sharing by Chih-Ching Thein,Ja-chen Lin.
[2] An overview of image Stenography by T.Morkel, J.H.P. Eloff, M.S.Olivier
[3] SteganPEG Stenography+JPEG by V.Lokeswara Reddy, Dr.A.Subramanyam, Dr.P.Chenna Reddy
[4] Some New methodologies for Image Hiding using Stenographic Techniques by Rajesh Kumar Tiwari and Gadadhar Sahoo