Cover as Key and Key as Data: An Inborn Stego

Siva Janakiraman, Jagannathan Chakravarthy, Badrinath Radhakrishnan, K. Thenmozhi, J.B.B. Rayappan and Rengarajan Amirtharajan
School of Electrical and Electronics Engineering, SASTRA University, 613401, India

Abstract: Communication playing the key role of daily lives has created revolutions since the Stone Age. In this fast-paced technological world, the utilization of information has gained towering momentum. Furthermore, information while transmitted over the internet with the swift escalation in the mass media and digital communication technology, the requirement for impregnable covert channel arises especially when critical information has to be communicated. Over the years, Information hiding techniques have evolved to conceal any critical data in cover files like images, audio and videos and thus, certifying the genuineness and secrecy of the transmitted data. Steganography is one such method where the very existence of the data in the cover image is hidden and thereby, provides for security by making the hidden data unconceivable. The cover file may include audio, video or image files. In this study, two methods have been proposed to embed image within an image with two and three layers of security, respectively. A decoder circuit is used in one of the layers to embed the data not only in the LSB bits but also in the other bits of the lower nibble. Moreover, two bits of data are embedded in a pixel by changing only one bit of the pixel. These methods show improved security without compromising on the PSNR and MSE values.

Key words: Information hiding, LSB substitution, spatial domain, decoder, steganography

INTRODUCTION

The versatility in the process of acquiring information has led to the question of authenticity of the transmitted information. In view of this, the necessity to secure the clandestine information from unauthorized users has become increasingly important. Various methodologies have been proposed by various authors to encrypt the data and the means of embedding it (Peticolas et al., 1999; Zaidan et al., 2010). Though impenetrable, these techniques are conceived to be mind-numbing and even constitute a substantial time delay with regard to encryption of the data, trailed by embedding and transmission and thereby, incumbent to be a novel way. Cryptography, steganography or watermarking serve as a medium for information security through an elucidated survey on numerous information hiding methodologies with its virtues, loss of credit and their classifications can be availed from Katzenbeisser and Peticolas (2000). Cryptography involves the jumbling of the undisclosed information providing privacy (Janakiraman et al., 2012a); while on the other hand, steganography seeks to hide the very existence of the communication itself (Hmood et al., 2010; Amirtharajan et al., 2012; Rabah, 2004; Provos and Honeyman, 2003; Rajagopalan et al., 2012). In this study, two embedding methods have been proposed to securely conceal the data inside a cover image and transmit it over a hostile channel. The decoder circuit which is used enables the user to embed two bits of data by changing only one grey level of the cover image thereby improving imperceptibility. The results obtained show that the PSNR and MSE values obtained are uncompromised while the security is enhanced.

LITERATURE REVIEW

Ali et al. (2011) proposed a secure steganography method to embed two bits of data by altering only one bit of the cover pixel. They used a decoder circuit to achieve this. The two bits of data were embedded by changing either the first, second or the fourth bit of the cover pixel. Similar approach was taken to embed three bits of data by changing a maximum of two bits of the cover pixel (Janakiraman et al., 2011). Extending this, a method to embed four bits of data by changing a maximum of two bits of the cover image was proposed (Janakiraman et al., 2012b). In all these methods, although the MSE and PSNR values were affected, the security was increased to a great extent as the data was not directly embedded in the LSBs of the image (Mielikainen, 2006;
Chan and Cheng, 2004). In these papers, the main objective was to increase the capacity of embedding while in this paper, the main focus is to increase the security by encrypting the data using bit operations. Various methods to embed data in spatial as well as frequency domain have been proposed (Amirtharaj J and Balaguru, 2009, 2011; Kumar et al., 2011; Janakiraman et al., 2012c). Many methods have been proposed to increase security have also been proposed by using encryption techniques and random embedding (Padmaa et al., 2011; Thani kaiselvan et al., 2011a, b). To escape from steganalysis attack, various methods have been proposed which employ embedding data not only in the LSB but also in other bits of the cover pixel (Lu et al., 2009; Qin et al., 2009a; Xia et al., 2009). Also, methods to detect the data using steganalysis attacks have been proposed (Qin et al., 2007, 2009b).

PROPOSED METHOD

Method 1: Pixel key for data chaining

Procedure for embedding: The data or message which has to be concealed and the cover image are first studied. Considering the secret data as a data image, a two bit XOR operation is performed between every two bit of the data image and 5th and 6th bit of the corresponding pixel of the cover image. The third bit of the particular cover pixel considered is checked. If it is 1, the two bit result obtained from the XOR operation is reversed i.e., the bit positions are interchanged; else if the third bit is 0, the result is kept unchanged. Now, the lower nibble of the cover pixel is sent to the decoder circuit. The decoder will give a two bit result. This is compared with the result of previous operation. When there is a mismatch, one of the bits in the bit positions 1, 2 or 4 is changed according to the lookup table; else the original cover pixel itself is considered as stego pixel. The embedding process is described with sample data in Table 3 for method-1. The embedding process has an advantage that only one bit in the cover pixel needs to be changed to embed two bits of data. This process is continued until all the data bits are completely get embedded in cover image. Similar steps are adopted to retrieve every two bit of the data form each stego pixel for the reconstruction of data image. Finally the resultant image is stored as the stego image.

Procedure for retrieval: For the retrieval procedure the stego image is examined. The lower nibble of the stego image is where the bits are embedded. Hence it is given to the decoder circuit to get a two bit result. Now, the third bit of the stego image is checked. This is the bit which determines the order of the bit stream so as to reconstruct the pixels for data image. Hence if this bit is 1, the decoder output bits are reversed i.e., the bit positions are interchanged to get back the bits in original form. As in the embedding stage XOR operation is performed with 5th and 6th bits of the stego cover pixel, to obtain data bits in its original form. The same procedure is carried out to extract the data bits from every pixel of the stego image until the reconstruction of full data image. The flowchart for embedding and retrieval are shown in Fig. 1 and 2, respectively.

Algorithm for embedding:

1. **Step 1:** Start
2. **Step 2:** Read the cover image and data image
3. **Step 3:** Perform XOR operation between every 2 bits of data image and 5th and 6th bits of corresponding cover pixel
4. **Step 4:** Interchange the bit positions of the 2 bit data obtained in step 3 if the 3rd bit of cover pixel is 1
5. **Step 5:** The lower nibble of cover image is given to decoder circuit
6. **Step 6:** The data obtained in step 4 is then compared with decoder output. If there is a mismatch then either 1st or 2nd or 4th bit of the cover pixel is altered according to lookup table
7. **Step 7:** The pixel obtained after step 6 is the stego pixel where two bits are embedded with a maximum of one grey level change in the cover pixel

Start

Read cover image and stego image

Perform XOR operation with secret data and 5th and 6th bits of cover pixel

3rd bit of cover image

If I

If 0

Apply lower nibble value of each cover pixel to decode

Decoder output - secret data

Yes

No

One bit of cover pixel is changed

Stego pixel

No change in cover pixel

If last pixel

Yes

Stop

Fig. 1: Method-1 flow chart for embedding

1970
Fig. 2: Method-1 flowchart for retrieval

Algorithm for retrieval:
Step 1: Start
Step 2: Read the stego image
Step 3: The lower nibble of stego image is given to the decoder circuit to get 2 bit output
Step 4: The data obtained from step 3 is reversed i.e., bit positions are interchanged if the 3rd bit of the stego pixel is 1
Step 5: Perform 2 bit XOR operation between data obtained from step 4 and 5th, 6th bits of the stego image
Step 6: Do steps 3 to 6 till all data bits are extracted
Step 7: Store the result as data image

Method 2: Key embedding and encrypted Hiding

Procedure for embedding: The cover image and the data image are obtained as the input from the user and kept separately. Then the 8-bit key for encrypting the data is also obtained from the user. The key is then logically XORed with each pixel of the data image before the start of embedding process. Then 2 bits of the encrypted data image are embedded in each cover pixel using the encoder circuitry and lookup table as described in the Method-1. However, in this method, the variable Di from the Table 3 has to be considered as the data image after encryption using the 8-bit key. The last four pixels of the cover image are reserved for embedding the 8-bit key which was used for encrypting the pixel values of the data image. The result is then stored as a stego image.

<table>
<thead>
<tr>
<th>Di</th>
<th>K</th>
<th>Ei (Di)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

D0: Data, Ei (Di): Encrypted data, Ei (Di) = Di/K, where, k is 5th and 6th bits of the cover pixel

Procedure for retrieval: The stego image is obtained as an input. The retrieval process uses the same procedure as described in Method-1 that uses the decoder circuitry and lookup table for extraction of data image which is in encrypted form. The key for decryption is obtained from the byte formed by accumulating the data bits from the last four stego pixels. The 8-bit key is then XOR-ed with each pixel of the encrypted data image retrieved from the stego cover to obtain the original data image. The flowchart for embedding and retrieval are shown in Fig. 3 and 4, respectively.

Algorithm for embedding:
Step 1: Start
Step 2: Read the cover image and data image
Step 3: Obtain the key from the user
Step 4: XOR the key with every data pixel
Step 5: All the encrypted data image pixels are embedded in the cover image as described in Method-1
Step 6: Embed the 8-bit key in the last four pixels of the cover image
Step 7: Store the resultant as stego image

Algorithm for retrieval:
Step 1: Start
Step 2: Read the stego image
Step 3: The data bits are retrieved from each stego pixel as described in Method-1 to obtain the data image in encrypted form
Step 4: The last byte of the retrieved data is taken as 8-bit key
Step 5: Ex-OR each retrieved byte with the 8-bit key to decrypt the data image
Step 6: Store the result as data image

The decoder circuit to which the lower nibble of each cover pixel is applied is shown in Fig. 5. The bits N2 and N3 are taken for comparison. The Table 1-3 show progressively how a data is embedded in a cover image using three layers of security.
**ERROR CALCULATIONS**

The quality of image after embedding the secret image inside the cover image can be calculated using different parameters such as Mean Square Error (MSE), Peak Signal to Noise ratio (PSNR) etc. They are calculated as follows:

\[
\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (X_{ij} - Y_{ij})^2
\]

where, \(X_{ij}\) is the pixel value of each pixel in Stego image and \(Y_{ij}\) is the pixel value of each pixel in the cover image.

M and N are the number of rows and number of columns of the cover image matrix. Lower the MSE value, lower is the difference between cover image and stego image.
where, $I_{max}$ is the maximum intensity value for a pixel present in the image. $I_{max}$ is 256 for 8 bit gray scale images. Higher PSNR value denotes good image quality i.e., there is less difference between the cover image and stego image.

**RESULTS**

In this study, four cover images Boat, House, Frog, and Hydrangeas of size 128*128 are taken as shown in the figures below. SASTRA logo is taken as the secret image which is of size 64*64 as shown in Fig. 6. Several experiments are carried out and the corresponding stego images are also shown. The results obtained using the two methods are also tabulated. The results are obtained with maximum embedding. The maximum embedding capacity is 25%. The cover images are shown in Fig. 7a-d. The stego images obtained using method 1 are shown in Fig. 8a-d. The stego images obtained using method 2 are shown in Fig. 9a-d, respectively. The results obtained are tabulated in Table 4.
Fig. 7(a-d): Cover images; (a) Boat, (b) Photographer, (c) Frog and (d) Hydrangeas

Fig. 8(a-d): Stego images using method 1; (a) Boat, (b) Photographer, (c) Frog and (d) Hydrangeas
CONCLUSION

This study presents two methods of encrypting and embedding data in an image such that the security and imperceptibility are increased. The first method uses three layers of security such as XOR-ing, bit position interchange and decoder. The second method uses two layers of security such as user key and decoder circuit. As the results show, these methods have comparable MSE and PSNR values with respect to the method proposed by Ali et al. (2011) but have higher levels of security. Therefore, it becomes much harder to crack using steganalysis attack. Also, the imperceptibility is increased as we do not embed directly in the LSBs but charge only one bit of the cover pixel to embed two bits of data. Thus, both the methods introduce several levels of security to embed the data in the image without compromising on the embedding capacity.

REFERENCES


