Rubik’s Cube: A Way for Random Image Steganography

Rengarajan Amirtharajan, M. Venkata Abhiram, G. Revathi, J. Bharathsimha Reddy, V. Thanikaiselvan and J.B.B. Rayappan

School of Electrical and Electronics Engineering, SASTRA University, 613401, India
IIM Shillong, Meghalaya, 793 014, India
School of Electronics Engineering, VIT University, Vellore-632014, Tamil Nadu, India

Corresponding Author: Rengarajan Amirtharajan, School of Electrical and Electronics Engineering, SASTRA University, 613401, India

ABSTRACT
This study proposes an innovative methodology that incorporates the famous Rubik’s cube into steganography. Rubik’s cube is a pivot mechanism that enables each face to turn independently thus allowing mixing up of colors. By using this mechanism to represent the three planes of a color image (Red, Green, and Blue) the randomness of the stego image can be improved. The pixel intensity values can additionally be used as pointers to indicate the type of shift that needs to be done to the pixels in three planes. This study contains the experimental results that validate the superiority of this methodology compared to other existing ones in terms of imperceptibility, robustness and with reasonable embedding capacity. It is also found to be more resistant to steganalysis.

Key words: Cryptography, data hiding, information security, Rubik’s cube, steganography

INTRODUCTION
Human way of communication has come a long way from cave drawings and drums to electronic mails. As more and more information is communicated electronically, new needs, issues and opportunities are born. Authenticity and secret sharing are becoming the prime concerns (Abdulfetah et al., 2010; Zaidan et al., 2011). Data encryption (Salem et al., 2011; Schneier, 2007) and data hiding (Amirtharajan and Rayappan, 2012a-d; Amirtharajan et al., 2012; Bender et al., 1996; Cheddad et al., 2010; Janakiraman et al., 2012a, b; Rajagopalan et al., 2012; Thanikaiselvan et al., 2011) are so increasingly being used to protect the sensitive data from disclosure when they are transmitted over an insecure channel.

Imperceptibility and high embedding capacity coupled with robustness are the three main criteria that decide the performance of any stego algorithm (Amirtharajan and Rayappan, 2012a-d; Amirtharajan et al., 2012). Imperceptibility and embedding capacity are the two main concerns of steganography while robustness is the prime concern of watermarking (Amirtharajan et al., 2012; Abdulfetah et al., 2010; Stefan and Fabin, 2000; Zeki et al., 2011).

Data can be hidden in a grayscale (Amirtharajan and Rayappan, 2012a) or color (Gutub, 2010; Padmaa et al., 2011) image because slight changes to colors of pixels are imperceptible to the eye.
(Amirtharaj and Rayappan, 2012a-d). The principle of LSB substitution can be used to induce these changes, that is to embed data into the cover image. The modified image is known as stego image (Chan and Cheng, 2004). Given a color image each pixel is represented by three bytes. Each byte represents a color component, typically red, green, and blue (Padmaa et al., 2011). Furtive information, in bits, is veiled by laying it as LSBs in pixel bytes. The least significant bit is either a 0 or 1. The former is also the same. Hence, typically no more than half LSBs get altered through bits shrouding.

An improvement to basic LSB substitution is to hide a bit in a pixel only if the pixel satisfies certain conditions (Zanganeh and Ibrahim, 2011) or after randomization of the pixel (Zhao and Lu, 2012). Hmood et al. (2010a, b) described the capacity of information hiding and an overview about steganography. Luo et al. (2011) explains secret sharing and data hiding for clandestine information adopting block truncation coding in compressed images. Several methods are available in the literature for different cover objects (Bender et al., 1996; Cheddad et al., 2010; Janakiraman et al., 2012a, b; Rajagopalan et al., 2012; Thenmozhi et al., 2012) like text (Al-Azawi and Fadhl, 2010; Xiang et al., 2011), image (Amirtharaj and Rayappan, 2012a-d; Amirtharaj et al., 2012; Luo et al., 2011; Mohammad et al., 2011), audio (Zhu et al., 2011) and video (Al-Frajat et al., 2010). Nevertheless a method is seldom considered for data hiding in scrambled images.

This study advises an unequalled model for image steganography in which Rubik’s cube theory is practised to scramble an image and then secret data entrenching is entitled using cyclic pixel indicator and LSB substitution.

PROPOSED METHOD

In this proposed methodology the pixels are first scrambled and then data is embedded based on their intensity values. For scrambling, the famous Rubik’s cube methodology is implemented (Yen and Lin, 2010). The three rows of a face of the Rubik’s cube are taken as the three components of the color pixel (Loukaoukha et al., 2012). For instance, the first row is taken as red, the second row green and third blue. Based on the third and fourth bits of the bytes of a particular component, the other two components are scrambled by rotation. Then by using the first and second bits of a channel as pointer, embedding is carried out in the other two channels. This accounts for improved randomness that can resist any steganalysis (Qin et al., 2010). Embedding capacity is also enhanced as all the three channels are effectively used in hiding data.

A comparison of the original image and the stego image would show an attacker which bits have been modified and may serve as the basis of a successful attack on the hidden data. That is why the original image should be destroyed right after the stego image has been prepared.

This study wishes for a methodology of \( k = 2 \) bit embedding derived from the expertise of Rubik’s cube. The algorithm comprises of both image scrambling and embedding to pull off the aspiration of secret sharing. Scrambling procedure is defined which involves shifting of bits. The diagram of Rubik’s cube and its rotated and scrambled image is shown in Fig. 1.

Pixel indicator technique is used in this paper where indicator channel is cyclically chosen. The proposed block diagram is given in Fig. 2. Here, the secret data is embedded in to cover image which is scrambled using Rubik’s cube. The embedded image is des scrambled as original image to get stego image. Then the stego image is scrambled using Rubik’s cube and then extracted to get the original data.
Fig. 1(a-c): (a) Rubik's cube (b) Rotated Rubik's cube and (c) Scrambled Rubik's cube

Fig. 2: Block diagram for proposed method

Algorithm for embedding:
- Read the cover image and secret data
- Divide the cover image into Red, Green and Blue planes
- Convert secret data into binary
- Read the number of rows and columns for scrambling using Rubik's cube
Algorithm for embedding: Continue

- Follow cyclic pixel indicator method (E.g. R is 1st plane, G is 2nd plane, B is 3rd plane) and consider 3rd and 4th LSBs in indicator channel for scrambling.
  - If the bits are:
    - 00 = don't scramble
    - 01 = scramble 2nd plane
    - 10 = scramble 3rd plane
    - 11 = scramble both the planes
  - Scrambling procedure:
    - In the first pixel of data channel, the first bit is temporarily assigned to a separate variable (Temp) and the subsequent bits are shifted one position right as in Fig. 3. After shifting nth bit to n-th position, the first bit is placed in the n-th position as in Fig. 4. This procedure is followed for scrambling in all the planes. For descrambling the image, the reverse procedure is adopted.
  - Merge all the scrambled planes to form a scrambled image and split that image into R, G and B planes again
  - Follow cyclic pixel indicator method and consider last 2 LSBs in the indicator channel for embedding
    - If the bits are 00, then don't embed
    - Else if 01, then embed 2 bits of secret data in 2nd plane
    - Else if 10, then embed 2 bits of secret data in 3rd plane
    - Else (bits = 11) embed 2 bits of secret data in both the planes
  - If all the secret data are embedded, descramble it as original image and then store it as resultant stego image

Algorithm for extraction:

- Read the stego image
- Split the image into red, green and blue planes
- Scramble the image by using 5th step from embedding algorithm
- Merge all the scrambled planes to form a scrambled image and split that image into R, G and B planes again
- Follow the cyclic pixel indicator method and consider last 2 LSBs in the indicator channel for extraction
  - If the bits are 01, then extract 2 bits from 2nd plane
  - Else if 10, then extract 2 bits from 3rd plane
  - Else (bits = 11) extract 2 bits from both the planes
- Store the original secret data

![Diagram of Algorithm](image)

**Fig. 3:** Move the first bit into the Temp

![Diagram of Algorithm](image)

**Fig. 4:** Return the Temp value to Array
The flow chart for embedding and extraction are given in Fig. 5 and 6, respectively. Here embedding and extraction process are explained in step by step manner.

RESULTS AND DISCUSSION
To justify the effectiveness of this study, the algorithm is simulated in MATLAB 7.1 with Lena, Baboon, Mahatma Gandhi and Temple as carrier images are depicted in Fig. 7a-10a with their corresponding stego images and accordingly their histograms are shown in Fig. 7b-10b. All images
Fig. 6: Flow chart for extraction

are of dimension 256×256×3 and approximately 30-35 kb in size. Here, pixel indicator technique is exploited to appreciate the goodness of the algorithm with the liberality in choosing the indicator for each embedding.

Tabulation for comparative results of MSE, PSNR, BPP and total number of bits embedded are tabulated in Table 1.

Thus, by fixing indicator channel, and based on that 2 bits are embedded in every iteration in each plane. As per the size and dimension of the images, MSE and PSNR values are vary in each
<table>
<thead>
<tr>
<th>Cover image</th>
<th>PI Methods</th>
<th>Channel red</th>
<th>Channel green</th>
<th>Channel blue</th>
<th>Bits per pixel</th>
<th>No of bits embedded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Proposed</td>
<td>0.4987</td>
<td>51.15</td>
<td>0.2594</td>
<td>53.9904</td>
<td>0.7033</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.227</td>
<td>47.24</td>
<td>1.3641</td>
<td>46.782</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.68</td>
<td>45.89</td>
<td>1.57</td>
<td>46.18</td>
<td>1.52</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2011)</td>
<td>1.2906</td>
<td>47.02</td>
<td>1.2374</td>
<td>47.2069</td>
<td>1.2049</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2012)</td>
<td>2.4387</td>
<td>44.25</td>
<td>2.3096</td>
<td>44.501</td>
<td>2.3389</td>
</tr>
<tr>
<td>Baboon</td>
<td>Proposed</td>
<td>0.4754</td>
<td>51.35</td>
<td>0.2587</td>
<td>54.0024</td>
<td>4.6733</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.065</td>
<td>42.04</td>
<td>4.002</td>
<td>42.108</td>
<td>4.2847</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2010)</td>
<td>2.61</td>
<td>43.96</td>
<td>2.65</td>
<td>43.88</td>
<td>2.72</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2011)</td>
<td>46.21</td>
<td>1.5544</td>
<td>46.2151</td>
<td>1.5904</td>
<td>46.1157</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2012)</td>
<td>2.3702</td>
<td>44.39</td>
<td>2.3255</td>
<td>44.467</td>
<td>2.3619</td>
</tr>
<tr>
<td>Mahatma Gandhi</td>
<td>Proposed</td>
<td>0.4576</td>
<td>51.25</td>
<td>0.2558</td>
<td>54.0512</td>
<td>3.8500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.348</td>
<td>46.83</td>
<td>2.4212</td>
<td>47.025</td>
<td>1.2478</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2010)</td>
<td>1.34</td>
<td>46.83</td>
<td>1.30</td>
<td>47.67</td>
<td>1.24</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2011)</td>
<td>42.28</td>
<td>3.2944</td>
<td>42.9530</td>
<td>3.1356</td>
<td>43.1677</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2012)</td>
<td>2.5278</td>
<td>44.03</td>
<td>0.5758</td>
<td>41.2404</td>
<td>2.3995</td>
</tr>
<tr>
<td>Temple</td>
<td>Proposed</td>
<td>0.4673</td>
<td>51.43</td>
<td>0.2569</td>
<td>54.0333</td>
<td>0.8280</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.835</td>
<td>45.45</td>
<td>1.766</td>
<td>45.682</td>
<td>1.632</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2010)</td>
<td>1.85</td>
<td>45.45</td>
<td>1.76</td>
<td>45.65</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2011)</td>
<td>47.85</td>
<td>1.1062</td>
<td>47.6924</td>
<td>1.1240</td>
<td>47.6322</td>
</tr>
<tr>
<td></td>
<td>Amirtharajan et al. (2012)</td>
<td>2.3143</td>
<td>44.49</td>
<td>2.3096</td>
<td>44.4967</td>
<td>2.3764</td>
</tr>
</tbody>
</table>

plane. If we take Lena image, Green plane has the highest PSNR of 53.9904 and minimum of 49.6593 in Blue plane. If all four images are likened, Temple plane offers relatively high PSNR in all of its 3 planes. Approximately, BPP witnessed in these images is found to be 0.6630 which is fairly straight for constant bit embedding. More or less 130677 bits are embedded in each plane which says about the efficiency of the algorithm.

In order to make the readers understand this libretto apart from analytical results, histograms and stego images are also portrayed. One can notice that covers and their corresponding stegos remain identical without giving a clue about secret sharing. As a result, they possess far above the ground imperceptibility which is what needed the most in a steganographic run. Moreover, encryption algorithm is designed by taking Rubik’s cube fundamentals. This would create more complexity along with security as well.

This study is justified in terms of steganalysis also to prove the performance. One such attack is Chi-square which generally tests the competence of any algorithm by giving pictorial representation. Here, for performance analysis sake, Mahatma Gandhi image is exposed to the test and graphical results are obtained in Fig. 11.

The two curves represent cover and stego images. Before 50 rows the probability of embedding gradually minimizes to zero that clears the fact that the stego seems as if it contains no secret hidden in it. However, more care is taken in bringing up the obscurity, which is revealed in the graph given. In general, probability of 1 is when all rows undergo embedding and 0 if none.
Fig. 7(a-b): (a) Lena cover image and its corresponding stego image and (b) Subsequent histograms for cover and stego images

Fig. 8(a-b): (a) Baboon cover image and its corresponding stego image and (b) Subsequent histograms for cover and stego images
Fig. 9(a-b): (a) Mahatma Gandhi cover image and its corresponding stego image and (b) Subsequent histograms for cover and stego images

Fig. 10(a-b): (a) Temple cover image and its corresponding stego image and (b) Subsequent histograms for cover and stego images
CONCLUSION

Rubik’s cube is a 3-D mishmash riddle, which has lent itself in discovering numerous theories and concepts in mathematics and engineering. This paper takes the idea of Rubik’s cube and formulates it in an image steganographic scheme. Secret data undergoes manipulation on some grounds of cryptography and further follows steganographic principles to share the secret. The experimental values show that this is one benevolent model for hiding secret in images. It also has the possibility of practical realization apart from owning secrecy and anonymity. The study is also tested against Chi-square to give good reason for. Thus, this study proposes a beneficial steganographic method when equated with previously available ones.

REFERENCES


