International Journal of Applied Engineering Research

Bit Pattern Selection Based Novel Method of Steganography in RGB Encoding Scheme Based Digital Images

Ghulam Gilanie*, Syeda Naila Batool, Aqsa Khursheed, Hina Shafique, Nimra Mahmood, Sana Cheema, Akkasha Latif, Muhammad Sajid, Muhammad Saeed

Department of Artificial Intelligence, Faculty of Computing, The Islamia University of Bahawalpur, Pakistan.

***Email addresses*: ghulam.gilanie@iub.edu.pk, nailashah313@gmail.com, aqsakhursheedbwp@gmail.com, hinach1912@gmail.com, nimramahmoodbwp@gmail.com, sanacheema887@gmail.com, akashacheema70@gmail.com, muhammad.sajid@iub.edu.pk, msaeed4771@gmail.com**

Abstract: We proposed a novel technique of steganography that hides information in images. When secret information is hidden in the image, there are less than 1.0% chances of change in image quality. We segregated literals of a message into 3, 3 and 2 bits and matched these bit patterns from red, green, and blue components of the cover image pixels. Hence, noting down these harmonized pixels, yields LOG table, which is encrypted using Rijndael Managed algorithm. We made an analysis between the reported method and other state-of-the-art steganographic techniques and found it successful.

Keywords: Steganography, LOG table, Rijndael Managed algorithm, Image Analysis

1.0 Introduction

Steganography is an art of hiding data in some media, preferably the digital media. Although, steganography is a prehistoric skill, however, computation makes it a powerful tool today [1]. Steganography enables us to obscure a secret message from the outside work rather than concealing the contents of a message like cryptography, hence more secluded technique than any other competitor manner of message hiding [2]. It changes to cover image to implant secret message to communicate in any standard format so that these changes are not detectable by foreign. Throughout history, people needed to communicate with each other, and now an ever-increasing number of people are using electronic means to send their messages. These same people are also becoming more aware of their right to privacy, and how many governments are introducing new laws to combat terrorism without thinking about the right to privacy of the average. Now, just encrypting emails is not sufficient, and with more computer viruses in circulation and the advent of spyware being used to steal personal information and documents from home computers, it has become necessary to ensure that a user's personal files are secure from attack. Encrypting individual files solved this problem but in doing so introduced another: if it's encrypted it must be worth encrypting [3-11].

Steganography is information hiding into some channel [12]. The channel may be audio, video, text, protocol, or an image. When information is embedded into some media/channel, the quality of source media is disturbed. In the case of an image, the image quality is also affected. There are some techniques available in literature to hide data into images, each one with its pros and cons. Even, different image formats have different methods of information hiding into them [13]. Different researchers are utilizing steganography techniques for some security purposes. There are two similar terms steganography and cryptography used in the way that they both protect important information. Steganography involves hiding information while cryptography encrypts the information [14, 15], while the proposed method deals only steganography. The proposed method is very novel in its nature that there is only less than 1% chance of change in image intensity values. The proposed method divides each token of a message into three identical literals, and then the patterns of literals are matched with the individual pixels of the image from each red, green, and blue portions. Where these three literals are matched, their position and length of a literal are packed to form a byte. Hence a table namely LOG table is generated. This generated LOG table is then encrypted using Rijndael managed encryption algorithm.

In this research work, we focused on novel image steganography techniques and compared the results with other traditional techniques reported for the same purpose. These include Least Significant Bit (LSB) and its modified versions namely LSB with a single character per two consecutive pixels, LSB with a random selection of pixels, and LSB with a character per two pixels with random selection. The histogram-based analysis shows that the proposed method is better than the traditional LSB methods. It causes minimum variations in the original color tone, which are impossible for the human eye to perceive and in many ways even undetectable during steg-analysis.

2.0 Literature review

l-Shatnawi et al., [16] proposed a novel approach of steganography, which searches identical bits between the message and the individual pixels of the image to hide text into these pixels. It has the main drawback that any hacker can detect the original text from the image with different combinations of bit patterns. Sadaf et al., [17] hide text in the randomly selected colored image of arbitrary size, using wavelet transform with the only drawback that the steg-analysis can easily detect the stego text from the cover image. Gutub et al., [18] introduced the concept of storing a variable number of bits in each channel of a pixel, based on the actual color values of that pixel. The main drawback of the method is the extra payload added with the actual pixel values resulting in a poor quality stego image. Khare et al., [19] designed a system that allows an average user to securely transfer text messages by hiding them in a digital image file using the local characteristics with in an image. However, the sender must calculate the local characteristics him/herself. Budhiraja et al., [20] studies invisible communication that usually deals with the ways of hiding the existence of the communicated messages. Parvez et al., [21] reported the RGB image based steganography method for storing a variable number of bits in each RGB channel based on the color value of each pixel. However, they changed the least significant bits of channels, consequently affecting the image quality. Bennett at al., reported that channels in which information can be hidden, and increasingly, reported sophisticated techniques to analyze and recover that information.

3.0 Proposed method

In RGB encoded color images, each pixel has at least three components, i.e., red, green, and blue. Each component is represented by a separate 8-bit value. So, the intensity value of each component can range from 0 to 255 only. Represented as decimal, for ease of human perception and reading, a value of (255, 0, 0) would describe a 100% red pixel (🟥). By mixing the contribution of each component, a large palette of colors can be represented.

Value mixtures such as (31, 187, 57) result in a shade of green (🟩), while (255, 255, 0) represents pure yellow (🟨). When any specific color is viewed closely, single digit modifications to the contribution level are imperceptible to the human eye i.e. a pixel with a value of (255, 255, 0) representing this yellow color (🟨) is indistinguishable from color with value of (254, 255, 0) representing a slightly different shade of yellow (🟨). No matter which technique is used for image steganography, image quality is at compromise. A decrease in image quality leads to detection during attack or steg analysis. To prevent even the minimum degradation of image quality, a novel method is proposed in this research work. It finds a pattern of the bit sequences of what is to be hidden in the original pixel values of the image. Hence, eliminating the need to alter the cover image in any way for almost all the character sequences of the secret message.

Let I be the cover image and T be the text to stego inside a cover image. For pattern matching, the bit sequences of each character of the T are grouped as 03 (three) MSBs, 03 (three) middle bits and 02 (two) LSBs, named as Lit1, Lit2 and Lit3 respectively. The pattern making process of a sample character 'A' is depicted in Table 1.

Table 1. Pattern making process of sample character 'A'

Character	ASCII Value	Binary	Literal$_1$ (Lit$_1$)	Literal$_2$ (Lit$_2$)	Literal$_3$ (Lit$_3$)
A	65	0100 0001	010	000	01

Upon having literals of T, these are matched with bit sequences of color components of each pixel of I. The process of pattern matching starts from MSB of the first pixel of the image. Sequentially searching for bit patterns of each literal. If any of the patterns is not matched with any of the color components, the next pixel of I is selected for pattern matching. The bit pattern matching process always starts from MSB, for instance Lit 1 is compared with 8th, 7th and 6th bit, if it is not a match then move on to the 7th, 6th and 5th bits and so on in red component. Lit 2 and Lit 3 are searched from green and blue components respectively.

If a pattern is matched successfully, the respective information is stored in a byte. This byte is named as Literal and post-fixed with a number according to the pattern whose information it holds, i.e., Literal1 for Patter1, Literal2 for Pattern 2 and Literal3 for Pattern 3 respectively. Each Literal stores the information as per format described in Table 2.

Table 2. Literal format

Literal (00-0000-00)(MSB to LSB)		
First 2 bits	**Next 4 Bits**	**Last 2 Bits**
The color component where the pattern was found. 00 for RED 01 for GREEN 10 for BLUE 11 *reserved for future use*	The starting bit position where the pattern was found. The maximum value that can be stored is ***1000b*** (8th bit), which is the MSB in a byte.	Number of bits in the pattern, either 2(10) or 3(11)

In the rarest case, if a pattern is not found in any of the pixel values of the image, the character byte is embedded into one of the pixels by the LSB method. But even for this, it is ensured that the most suitable pixel is selected to modify, i.e., the pixel whose bit sequences match at least two of the patterns while the third one is embedded into the color component from which neither of the two other patterns was found. In case a pixel is modified, we also need to know and keep track of it, where and how it was modified, i.e., which color component was modified, and the number of bits that were modified. This requires a byte, whose 1st three bits (from MSB to LSB) show the color component (100 for Red, 010 for Green and 001 for Blue) and the last 2 bits indicate the number of bits modified. Furthermore, it will be ensured that this pixel is not modified in future for similar cases of
unsuccessful matching.

The data structure used in this proposed method to store the information of character patterns is a LOG table containing two integers and 4 bytes in each row. The first integer holds the value of the character index of T, the second integer holds the pixel number of the I, the next three bytes contain the three literals respectively, while the last byte indicates whether the pixel was modified or not.

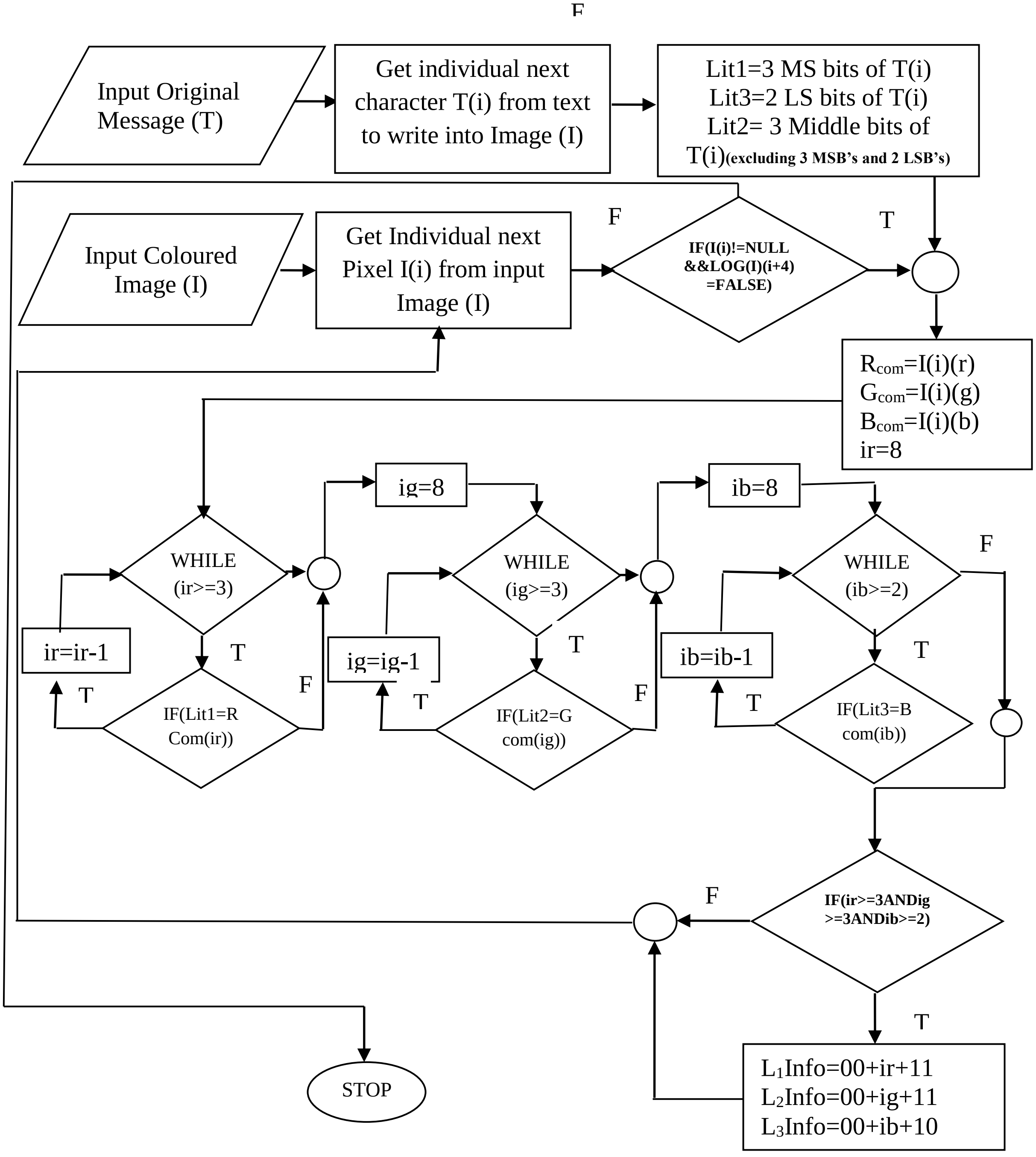
When this process is completed, and a LOG is generated, it is then encrypted using the Rijndael Managed encryption algorithm. The cover image along with encrypted LOG table is communicated to the destined location. This extraction of secret message from the cover image and sending them separately exploits the dependency of the LOG table on the cover image and vice versa; making it a challenge for attackers to extract the secret message when they have only one of the objects in hand. The flowchart of the proposed algorithm is depicted in Figure 1.

Algorithm

```
Step1.   [Initialize] I= Image
Step2.   [Initialize] T= Text
```

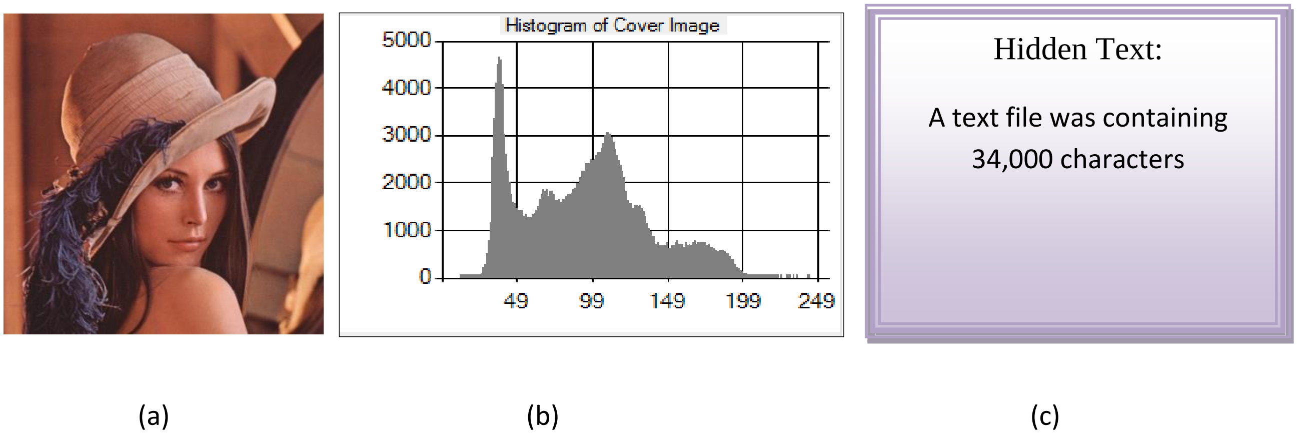
```
Step3.  [Initialize] Counter=1
Step4.  WHILE (T(i)! =NULL) DO
a.  Lit1=3 MSB of T(i)
b.  Lit2=2 LSB of T(i)
c.  Lit3=3 Middle bits of T(i) (excluding Lit1 and Lit2)
d.  WHILE (I(i)! = NULL && LOG (I)(i+4) =FALSE) DO
    a.  Rcom=I(i)(r)     //Red Component of Pixel I(i)
    b.  Gcom=I(i)(g)     //Green Component of Pixel I(i)
    c.  BCom=I(i)(b)     //Blue Component of Pixel I(i)
    d.  [Initialize] ir=8
    e.  WHILE (ir>=3) DO
        i.  IF (Lit1= Rcom(ir) THEN
            1.  Break.
        ii.  ELSE
            1.  ir=ir-1
        iii.  END IF
    f.  END WHILE
    g.  [Initialize] ig=8
    h.  WHILE (ig>=3) DO
        i.  IF (Lit2= Gcom(ig) THEN
            1.  Break.
        ii.  ELSE
            1.  ig=ig-1
        iii.  END IF
    i.  END WHILE
    j.  [Initialize] ib=8
    k.  WHILE (ib>=2) DO
        i.  IF (Lit3= Bcom(ib) THEN
            1.  Break.
        ii.  ELSE
            1.  ib=ib-1
        iii.  END IF
    l.  END WHILE
e.  IF (ir>=3 && ig>=3 && ib>=2) THEN
    a.  Lit1Info=00+ir+11
    b.  Lit2Info=01+ig+11
    c.  Lit3Info=11+ib+10
    d.  LOG (I)(i) =Counter
    e.  LOG(I)(i+1) =Lit1Info
    f.  LOG(I)(i+2) =Lit2Info
    g.  LOG(I)(i+3) =Lit3Info
    h.  LOG(I)(i+4) =FALSE
f.  END IF
```

```
        a.  I(i)=I(i)+i
        b.  Counter=Counter+1
    g.  END WHILE
        a.  T(i)=T(i)+1
    h.  END WHILE
Step5.  STOP
Step6.  END Algorithm
```



4.0

For research and experimental purposes, images of different categories of size 256×256 were used. Figure 2 (a) shows the cover bitmap Lena image. Figure 2 (b) shows its histogram and Figure 2(c) shows a “TEXT” file containing 34000 characters, which were used to embed into cover image. Figure 3 (a) shows the image after embedding the data into it by the sequential pixel selection and LSB embedding method. Figure 3 (b) shows the histogram of Figure 3(a). It is visually perceivable from Figure 2 (a) and Figure 3 (a) and their respective histograms shown in Figure 2 (b), and Figure 3(b) that there is minor difference in image which is not easily interpretable by the human eye. However, there is some change in image quality after embedding text into it. To verify the extent of change in image after steganography, we calculated the difference between them and visualized the difference in Figure 3 (c) represented by red lines. This is the actual change occurred in cover image after steganography using sequential ordering of storing data into pixels on LSB one character per pixel. Data was embedded into an image with the format as 3 bits at RED component, 3 bits at GREEN component and 2 bits of BLUE component of each sequentially selected pixel.

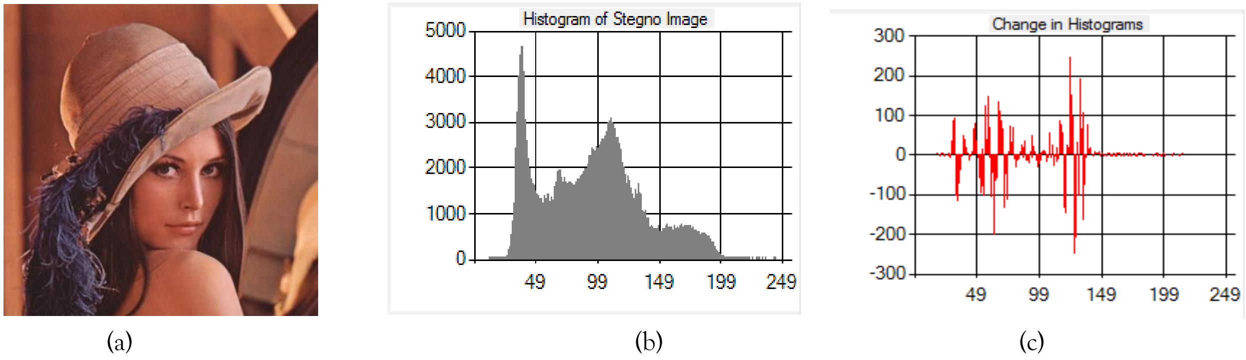


Hidden Text:

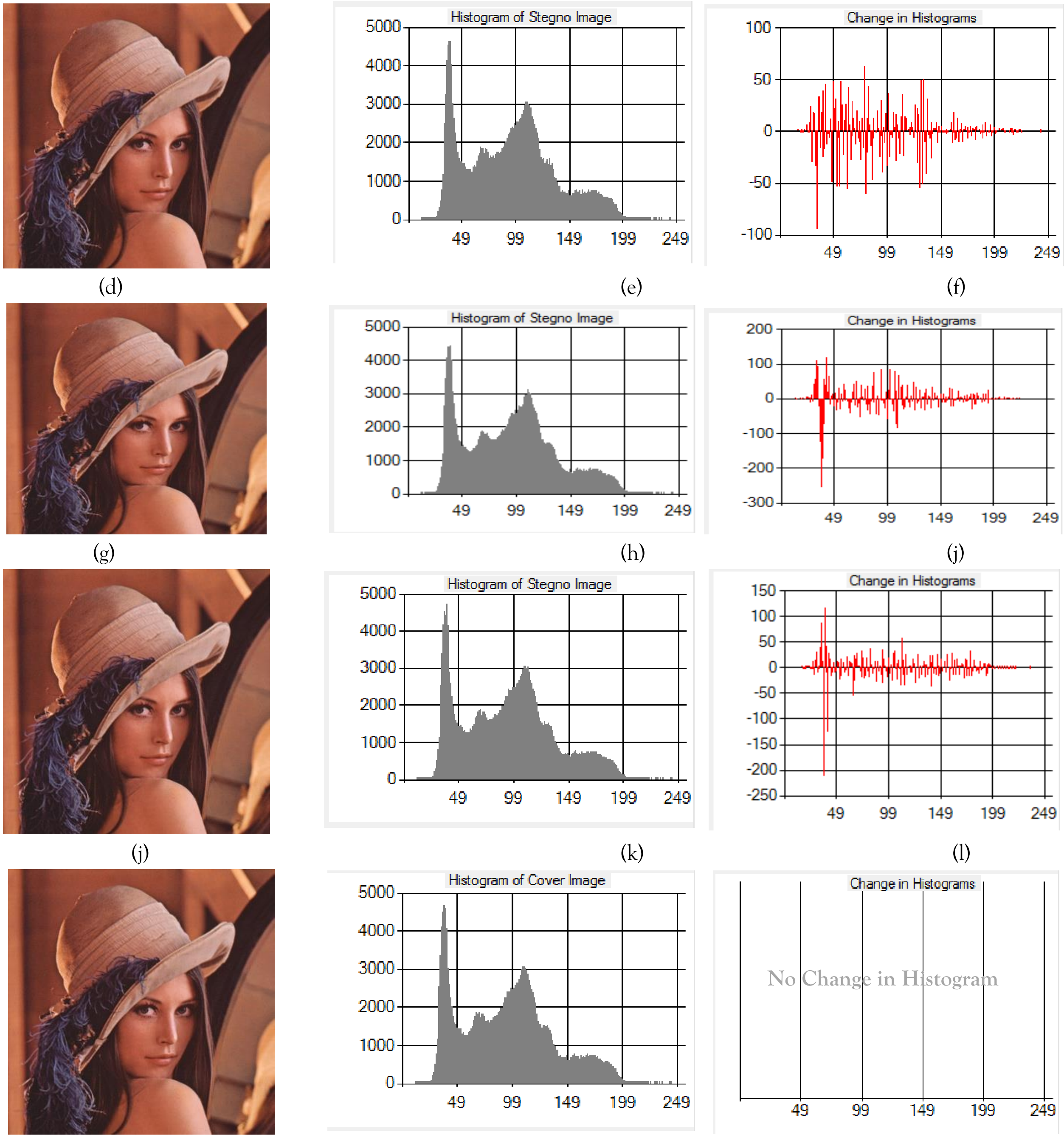
A text file was containing 34,000 characters

(a) (b) (c)

Figure 2 (a) Original BMP Portrait Image (Lena) (b) Histogram of the image shown in Figure 2(a) (c) The text file to hidden in the image.



(a) (b) (c)

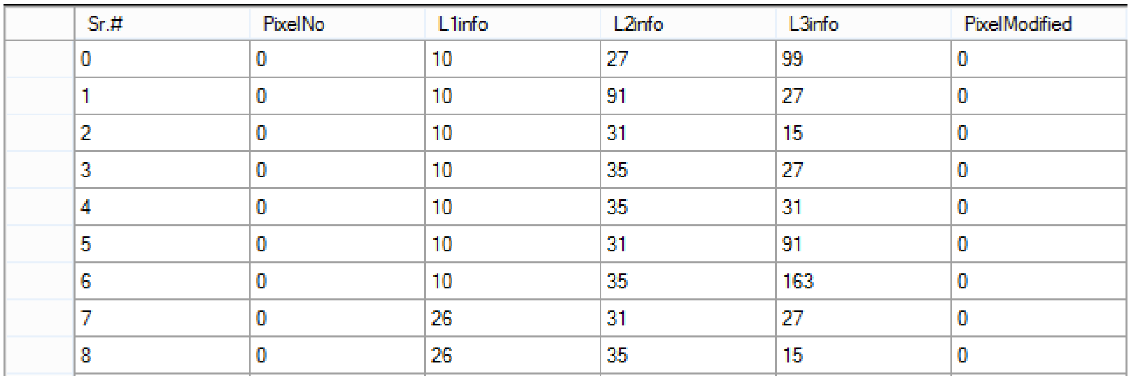


(d) (e) (f)

(g) (h) (j)

(j) (k) (l)

(m) (n) (o)



Sr.#	PixelNo	L1info	L2info	L3info	PixelModified
0	0	10	27	99	0
1	0	10	91	27	0
2	0	10	31	15	0
3	0	10	35	27	0
4	0	10	35	31	0
5	0	10	31	91	0
6	0	10	35	163	0
7	0	26	31	27	0
8	0	26	35	15	0

(p)

Figure 3 (a) stego image after LSB sequential (b) histogram of image shown in 3 (a), (c) histogram showing the change between the image shown in 2(b) and 3(b), (d) stego image after LSB sequential one char/ two pixels (e) histogram of image shown in 3 (d), (f) histogram showing the change between 2(b) and 3(e), (g) stego image after LSB modification on random pixel selection (h) histogram of image shown in 3 (g), (i) histogram showing the change between the histograms shown in 2(b) and 3(h), (j) stego image after LSB random one char/two pixels (k) histogram of image shown in 3 (j), (l) histogram showing the change observed in 2(b) and 3(j) , (m) stego image after proposed algorithm (n) histogram of image shown in 3(m), (o) histogram showing the change the in histograms between 2(b) and 3(m), (p) the LOG table generated after applying proposed method of steganography on image shown in 2(a)

Figure 3 (d) shows the image after embedding the text message into it with sequential pixel selection where each character byte was embedded on each two consecutive pixels. Figure 3 (e) shows the histogram of Figure 3 (d). It is also visually perceivable from the images shows in Figure 2 (a) and Figure 3 (a) and their histograms are shown in Figure 2 (b) and Figure 3 (b) respectively that there is minor change in the image quality, which is not easily detectable by naked human eye. Therefore, to highlight the difference between these two ones, we calculate the difference of histograms as shown in Figure 3 (f) shown in red lines. Hence little changes are also observed in the image after embedding text using sequential pixel selection of one character at two consecutive pixels. These were the changes, occurred in the cover image after applying steganography using the LSB embedding methods with sequential pixel selection, while each character byte is divided upon two consecutive pixels, where one bit modifies the LSB of RED component; next bit modifies the LSB of GREEN component, next two bits modify the BLUE component of the 1st pixel. Hence, the same pattern modifies the color components of the 2nd pixel in the same order.

Figure 3 (g) shows the image obtained after steganography when random pixels are selected to embed text while Figure 3 (h) shows the histogram of figure shown in Figure 3 (g). Here the difference between the cover and stego image though can't be perceived, so, their difference is shown in Figure 3 (i). From Figure 3 (i), it is observed that the change in stego image is identical to the stego image shown in Figure 3 (c), but the change occurred in the stego image globally due to random selection of pixels.

Figure 3 (j) shows resultant stego image obtained after steganography with method like that used in Figure 3 (d) with the only difference that the pixels are selected randomly. Figure 3 (k) shows the histogram of the figure shown in Figure 3 (j), while Figure 3 (l) shows the difference between the histograms values of Figure 3 (b) and Figure 3 (k). The observed changes are almost equal to the changes, occurred in single sequential character per two pixels. However, due to random pixel selection, the changes are dispersed in the image. Conclusively, these previously described data embedding techniques, first, are similar and secondly, cause changes in image quality, which is easily detectable by any of the automated systems.

Figure 3 (m) shows the stego image obtained through the proposed novel method of steganography after embedding the same text file containing 34000 characters. Figure 3 (n) shows the histogram of the stego image shown in Figure 3 (m). It is verified from Figure 3(o) that the proposed method doesn't modify the pixels of the cover image. Resultantly, no change occurred even in a single pixel of the cover image. Rather, a pattern selection technique is applied for extracting the bit pattern of characters from the byte values of different color components of the pixel. The extracted information is saved into a LOG table as shown in Figure 3 (p).

5.0 Conclusion

In this research work, we proposed and implemented a novel method of steganography, which is so efficient that there is less than 1% chance of change in image quality. The proposed method generated appreciative results when applied to different types of images. The proposed method partitioned the literal of text into three literals and searched for their patterns in pixel components. After successful matching of text message character bits, the findings are packed into bytes and are stored in a LOG table. The LOG table is then encrypted using Rijndael Managed encryption algorithm for added security when sent on communication channel. Experiments showed that more than 99% of images already contain literal patterns, eliminating the need to modify any on the image bit to hide information from it. LOG table works as a secret key, hence no need for a further key to generate and to transfer along with cover image.

References

[1] M. Attique *et al.*, "Colorization and Automated Segmentation of Human T2 MR Brain Images for Characterization of Soft Tissues," *PLoS ONE,* doi:10.1371/journal.pone.0033616 vol. 7, no. 3, p. e33616, 2012.

[2] G. Gilanie, M. Attique, U. Hafeez, S. Naweed, E. Ahmed, and M. Ikram, "Object extraction from T2 weighted brain MR image using histogram based gradient calculation," *Pattern Recognition Letters,* vol. 34, no. 12, pp. 1356-1363, 2013.

[3] A. Anderson, "Steganography: an Inside Look at Hiding Messages and Data," University of Exeter, 2006.

[4] G. Gilanie, U. I. Bajwa, M. M. Waraich, and M. W. Anwar, "Risk-free WHO grading of astrocytoma using convolutional neural networks from MRI images," *Multimedia Tools and Applications,* vol. 80, pp. 4295-4306, 2021.

[5] G. Gilanie *et al.*, "Coronavirus (COVID-19) detection from chest radiology images using convolutional neural networks," *Biomedical Signal Processing and Control,* vol. 66, p. 102490, 2021.

[6] G. Gilanie, U. I. Bajwa, M. M. Waraich, and Z. Habib, "Automated and reliable brain radiology with texture analysis of magnetic resonance imaging and cross datasets validation," *International Journal of Imaging Systems and Technology,* vol. 29, no. 4, pp. 531-538, 2019.

[7] G. Gilanie, N. Nasir, U. I. Bajwa, and H. Ullah, "RiceNet: convolutional neural networks-based model to classify Pakistani grown rice seed types," *Multimedia Systems,* pp. 1-9, 2021.

[8] G. Gilanie, H. Ullah, M. Mahmood, U. I. Bajwa, and Z. Habib, "Colored Representation of Brain Gray Scale MRI Images to potentially underscore the variability and sensitivity of images," *Current Medical Imaging,* vol. 14, no. 4, pp. 555-560, 2018.

[9] K. Asghar, G. Gilanie, M. Saddique, and Z. Habib, "Automatic enhancement of digital images using cubic Bézier curve and Fourier transformation," *Malaysian Journal of Computer Science,* vol. 30, no. 4, pp. 300-310, 2017.

[10] G. Gilanie, U. I. Bajwa, M. M. Waraich, and Z. Habib, "Computer aided diagnosis of brain abnormalities using texture analysis of MRI images," *International Journal of Imaging Systems and Technology,* vol. 29, no. 3, pp. 260-271, 2019.

[11] M. J. Iqbal, U. I. Bajwa, G. Gilanie, M. A. Iftikhar, and M. W. Anwar, "Automatic brain tumor segmentation from magnetic resonance images using superpixel-based approach," *Multimedia Tools and Applications,* vol. 81, no. 27, pp. 38409-38427, 2022.

[12] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *Computing Practices,* vol. 1, no. 1, pp. 26-34, 1998.

[13] S. K. Bandyopadhyay, D. Bhattacharyya, D. Ganguly, S. Mukherjee, and P. Das, "A Tutorial Review on Steganography," *UFL & JIITU,* vol. 1, no. 1, pp. 105-114, 2008.

[14] A Joseph Raphael and V. Sundaram, "Cryptography and Steganography – A Survey," *International Journal of Computer technology and applications,* vol. 2, no. 3, pp. 626-630, 2011.

[15] Khalil Challita and H. Farhat, "Combining Steganography and Cryptography: New Directions," *International Journal on New Computer Architectures and Their Applications (IJNCAA),* vol. 1, no. 1, pp. 199-208, 2011.

[16] A. M. Al-Shatnawi, "A New Method in Image Steganography with Improved Image Quality," *Applied Mathematical Sciences,* vol. 6, no. 79, pp. 3907-3915, 2012.

[17] D. M. Y. Saddaf Rubab, "Improved Image Steganography Technique for Colored Images using Huffman Encoding with Symlet Wavelets," *International Journal of Computer Science Issues,* vol. 9, no. 2, pp. 1694-0814, 2012.

[18] M. T. Parvez and A. A. A. Gutub, "RGB Intensity Based Variable-Bits Image Steganography," in *Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE*, 2008, pp. 1322-1327.

[19] Akhil Khare , Meenu Kumari, and P. Khare, "Efficient algorithm for digital image steganography " *Journal of information, knowledge and research in computer science and applications,* vol. 1, no. 1, pp. 1-5, 2014.

[20] B. Dunbar, "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment," *SANS Institute InfoSec Reading Room,* vol. 1, no. 1, pp. 1-11, 1/18/2002.

[21] Shikha Sharda and S. Budhiraja, "Image Steganography: A Review," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, no. 1, pp. 707-710, 2013.