BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 1
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0006

Digital Image Steganography Using Bit Flipping

Aditya Kumar Sahu[1,2]*, Gandharba Swain*[1]*, E. Suresh Babu*[1]

[1]*Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, 522502 Andhra Pradesh, India*
[2]*Department of Computer Science & Engineering, GMRIT, Rajam, 532127 Andhra Pradesh, India*
E-mails: adityasahu.cse@gmail.com gswain1234@gmail.com sureshbabu.erukala@kluniversity.in

Abstract**:** *This article proposes bit flipping method to conceal secret data in the original image. Here a block consists of 2 pixels and thereby flipping one or two LSBs of the pixels to hide secret information in it. It exists in two variants. Variant-1 and Variant-2 both use 7th and 8th bit of a pixel to conceal the secret data. Variant-1 hides 3 bits per a pair of pixels and the Variant-2 hides 4 bits per a pair of pixels. Our proposed method notably raises the capacity as well as bits per pixel that can be hidden in the image compared to existing bit flipping method. The image steganographic parameters such as, Peak Signal to Noise Ratio* (*PSNR*)*, hiding capacity, and the Quality Index* (*Q.I*) *of the proposed techniques has been compared with the results of the existing bit flipping technique and some of the state of art article.*

Keywords**:** *Steganography, Least Significant Bit* (*LSB*) *substitution, bit flipping, capacity.*

1. Introduction

Information hiding became a dominant area in every aspect of life today. The real threat to data is in the field of digital data communication. The Internet is the principal entity for carrying the digital data. The security to the data is a matter of real interest. In this aspect, cryptography and steganography are the prominent fields for achieving surveillance to secret data. Cryptography misleads the unapproved entity by manipulating the original information [1-3]. Steganography, other side shields the data so that the unapproved entity cannot even predict the survival of data. It comes in diverse ways such as implanting the data in the image, video, audio, etc., [4, 28]. Image steganography has gained the attention of many researchers during the years. Implanting text data in the original or cover image is image steganography; the new image which carries the information is resulting image [3].

The efficacy of any image steganographic method depends on various parameters. Parameters like capacity, Bits Per Pixel (BPP), Peak Signal to Noise Ratio (PSNR), Quality Index (Q.I) determines the impression of the method [2-5]. The amount of information a resulting image can hide is the capacity for the method. The bits per pixel (bpp) are the measure of the number of secret data in bits that are concealed in the pixel of an original image. The PSNR is a metric for finding the quality of a resulting image. The more the PSNR is the better the method Q.I. tells about the closeness between the original image and resulting image [5].

Least Significant-Bit (LSB) is the familiar and simpler technique that gives better capacity but also the relative reduction in the quality of resulting image [6]. The use of Moderately-Significant Bit (MSB) by pixel adjustment process of the cover image for concealing information has been suggested by R. Wang, C. Lin and J. Lin [7]. A genetic algorithm based on perceptual modelling with a combination of rightmost LSB in order to achieve large capacity has been proposed in [8]. Although the capacity increases by using R. Wang, C. Lin and J. Lin [8] method it's price is a hike in the complexity. To avoid the scenario and to reduce the complexity Chan and Cheng [9] suggested an optimal pixel adjustment process. The LSB bit is not altered straightforward by which the error can be cut down drastically.

The combination of cryptography and steganography can achieve a better security for the data [10]. LSB methods are valuable for hiding the larger magnitude of data. But, Fridrich, Goljan and Du [11] found the weakness of LSB methods. In 2003, Wu and Tsai [12] suggested a unique method on the ground of image steganography which is known as Pixel Value Differencing (PVD). Here the difference between two consecutive pixels is taken into account and a new difference value is found to concealing the secret data. The amount of secret data that will be hidden is dependent upon the difference value between the consecutive pixel. A larger difference value indicates a higher amount of secret data hidden in the image. The image can be segregated into either smooth area or edge area. The edge areas can be utilized to hide large number of data compared to smooth areas. A large variety of diversified techniques in combination with PVD and LSB has been suggested in the literature [13-23]. Swain [24] has given a method for inserting secret information by using Group of Bits Substitution (GBS). He suggested 1GBS and 2GBS techniques. The 1GBS technique hide 1 bit and 2GBS hides 2 bits of secret data. A novel track in image steganography called LSB array has been suggested in [25]. The above works have been further continued in [26-27].

2. Related work

The LSB substitution directly replaces the LSB bits of the pixel. Although the LSB technique offers good capacity but they are exposed to steganalytic analysis like RS-steganalysis [23]. In 2016, Kumar and Chand [2] proposed a reversible bit flipping method for concealing 2 bits of secret data in a segment of 2 pixels. In each segment the last LSB bits of the pixels have been exploited for concealing the secret data at the first layer of embedding. In the existing work, a segment of 2 pixels hides only 2 bits of secret information and again it does not directly substitute the LSB bits. Hence the chance of attacks is avoided. This motivates us to propose an improved bit flipping method to further improve the embedding capacity. This paper proposes two variants of bit flipping which can conceal 3 and 4 bits of secret data respectively in a segment of two pixels. The embedding and extraction process of the proposed variants are discussed below.

3. Proposed bit flipping technique

The bit flipping technique exists in two variants, (i) Bit flipping-1, and (ii) Bit flipping-2. Both techniques divide the cover image into blocks. Each block (S) consists of 2 pixels say, S_x and S_{x+1}, x=1 to N – 1, where N is the total number of pixel elements of the image.

3.1. Bit flipping-1

In this, the 7th and 8th bits of the pixels are utilized for hiding the secret information. Each block (S) hides 3 bits of secret data sequentially. The embedding and extraction procedures are outlined below.

3.1.1. Embedding algorithm

There are six steps.

Step 1. Change the data to be hidden into binary. The dimension of secret data is also changed to 18 bit. This is placed at the top of the binary message. The message is now combination of both.

Step 2. Read the original image (I_x). Convert it to binary also read the last 2 bits, i.e., 7th and 8th LSB bits to form the location map from all the pixels of the original image I_x, where, x=1 to N, and each I_x is a pixel of 8-bit length. Compress the location map and send it to the receiver.

Step 3. Read the secret data in binary. Divide the secret data into a block of three bits length. So, the three bits can range from 000 to 111.

Step 4. Let the block of secret data is k_i, where i = 1 to n/3, where n is the length of the secret message.

Step 5. For each block (S) of the original image (I_x), let S_x and S_{x+1} be the two consecutive pixels of the block. For each block (S) of original image (I),

If k_i= 000, No change to any of the pixels of the block.
Else if k_i= 001, Flip the 8th LSB bit of S_x only.
Else if k_i= 010, Flip the 7th LSB bit of S_x only.
Else if k_i= 011, Flip the 8th LSB bit of S_{x+1} only.
Else if k_i= 100, Flip the 7th LSB bit of S_{x+1} only.
Else if k_i= 101, Flip the 7th, 8th LSB bits of S_x only.
Else if k_i= 110, Flip the 7th, 8th LSB bits of S_{x+1} only.
Else if k_i= 111, Flip the 7th, 8th LSB bits of both the block.

Step 6. Transmit the obtained resulting image (G) along with the compressed location map to the receiver. The embedding process is successful.

3.1.2. Extraction algorithm

The extraction algorithm is opposite of embedding. The various steps for extracting the secret data are as follows.

Step 1. Decompress the compressed location map and find the 7th and 8th bit of original image. Initialize M_i to empty and initialize the counter, count = 1.

Step 2. For each block (*S*) of the resulting image (*G*) repeat Step 3.

Step 3. Compare 7th and 8th bits of each block (*S*) of resulting image with the location map.

If 7th and 8th LSB bits of both pixels S_x and S_{x+1} are same as corresponding bits of location map then extract 000 and concatenate to M_i .

Else if only 8th LSB bit of S_x has changed then extract 001 and concatenate to M_i .

Else if only 7th LSB bit of S_x has changed then extract 010 and concatenate to M_i .

Else if only 8th LSB bit of S_{x+1} has changed then extract 011 and concatenate to M_i .

Else if only 7th LSB bit of S_{x+1} has changed then extract 100 and concatenate to M_i .

Else if only 7th, 8th LSB bits of S_x has changed then extract 101 and concatenate to M_i .

Else if only 7th, 8th LSB bits of S_x and S_{x+1} changed then extract 110 and concatenate to M_i .

If 7th and 8th bits of both pixels S_x and S_{x+1} have changed then extract 111 and concatenate to M_i .

Step 4. Set count = count + 3. If count < 18 then go to Step 3, otherwise go to Step 5.

Step 5. Convert the 18 bits in M_i to decimal and then multiply by 7, which is the length of the embedded message in bits, let it be n.

Step 6. Reinitialize M_i to zero and for *i* = 1 to $(n - 18)/3$ repeat Step 3. Then we get M_i with *n* – 18 bits length.

Step 7. Convert the bits of M_i to characters. The extraction is successful.

3.2. Bit flipping-2

The7th and 8th bits of the original image (I_x) are used to hide the secret data. Each block (*S*) hides 4 bit of secret data sequentially. The embedding and extraction procedures are as follows.

3.2.1. Embedding algorithm

Step 1. Change the data to be hidden into binary. The dimension of secret data is also changed to 18 bit. This is placed at the top of the binary message. The message is now combination of both.

Step 2. Read the original image (I_x). Convert it to binary. Where, *x*=1 to *N*, where each I_x is a pixel of 8 bit length. Read the last 2 bits that is 7th and 8th bit from all the pixels of the original image (I_x) and form the location map. Compress the location map and send it to the receiver.

Step 3. Read the secret data in binary. Convert it into blocks of four bits in length. The four bits can range from 0000 to 1111.

Step 4. Let the block of secret data is k_i where i = 1 to n/4, where n is the length of the secret message.

Step 5. For each block (S) of original image (I), let S_x and S_{x+1} be the two consecutive pixels of the blocks.

If k_i = 0000, No change to any of the pixels of the block.

Else if k_i= 0001, Flip the 8th LSB bit of S_{x+1} only.

Else if k_i= 0010, Flip the 7th LSB bit of S_{x+1} only.

Else if k_i= 0011, Flip the7th, 8th LSB bits of S_{x+1} only.

Else if k_i= 0100, Flip the 8th LSB bit of S_x only.

Else if k_i= 0101, Flip the 8th LSB bits of both S_x and S_{x+1}.

Else if k_i= 0110, Flip the 8th LSB bit of both S_x and 7th LSB bit of S_{x+1}.

Else if k_i= 0111, Flip the 8th bit of S_x and 7th and 8th LSB bits of S_x and S_{x+1}.

Else if k_i= 1000, Flip the 7th LSB bit of S_x only.

Else if k_i= 1001, Flip the 7th LSB bit of S_x and 8th LSB bit of S_{x+1}.

Else if k_i= 1010, Flip the 7th LSB bits of both S_x and S_{x+1}.

Else if k_i= 1011, Flip the 7th LSB bit of S_x and both 7th and 8th LSB bits of S_{x+1}.

Else if k_i= 1100, Flip the 7th, 8th LSB bits of S_x only.

Else if k_i= 1101, Flip the 7th, 8th LSB bits of S_x and 8th LSB of S_{x+1}.

Else if k_i= 1110, Flip both the 7th, 8th LSB bits of S_x and 7th LSB bit of S_{x+1}.

Else if k_i= 1111, Flip the 7th, 8th LSB bits of S_x and S_{x+1} .

Step 6. Transmit the obtained resulting image (G) along with the compressed location map to the receiver. The embedding process is successful.

3.2.2. Extraction algorithm

The extraction algorithm is opposite of embedding process. The various steps of retrieving the secret data are outlined below.

Step 1. Decompress the compressed location map and find the 7th and 8th bits of the original image. Initialize M_i to empty and initialize the counter, count=1.

Step 2. For each block (S) of the resulting image (G), repeat the Step 3

Step 3. Compare each block (S) of resulting image with the location map.

If 7th and 8th LSB bits of both pixels S_x and S_{x+1} are same as corresponding bits of location map then extract 0000 and concatenate to M_i.

Else if only 8th LSB bit of S_{x+1} has changed then extract 0001 and concatenate to M_i.

Else if only 7th LSB bit of S_{x+1} has changed then extract 0010 and concatenate to M_i.

Else if only 7th, 8th LSB bits of S_{x+1} has changed then extract 0011 and concatenate to M_i.

Else if only 8th LSB bit of S_x has changed then extract 0100 and concatenate M_i.

Else if 8th LSB bit of S_x, S_{x+1} both have changed then extract 0101 and concatenate to M_i.

Else if 8th LSB bit of S_x and 7th LSB bit of S_{x+1} both has changed then extract 0110 and concatenate to M_i.

Else if 8th LSB bit of S_x and 7th, 8th LSB bits of S_{x+1} . both has changed then extract 0111 and concatenate to M_i.

Else if only 7th LSB bit of S_x has changed then extract 1000 and concatenate to M_i.

Else if 7th LSB bit of S_x and 8th LSB bit of S_{x+1} has changed then extract 1001 and concatenate to M_i.

Else if 7th LSB bit of S_x and S_{x+1} both has changed then extract 1010 and concatenate to M_i.

Else if 7th LSB bit of S_x and 7th and 8th LSB bits of S_{x+1} both have changed then extract 1011 and concatenate to M_i.

Else if 7th, 8th LSB bits of S_x has changed then extract 1100 and concatenate to M_i.

Else if 7th, 8th LSB bits of S_x and 8th LSB bit of S_{x+1} both have changed then extract 1101 and concatenate to M_i.

Else if 7th, 8th LSB bits of S_x and 7th LSB bit of S_{x+1} both have changed then extract 1110 and concatenate to M_i.

Else if 7th, 8th LSB bits of S_x and S_{x+1} both have changed then extract 1111 and concatenate to M_i.

Step 4. Set count = count + 4. If count ≤ 18 then go to Step 3, otherwise move to Step 5.

Step 5. Decimalize the 18 bits in M_i and multiply by 7, which is the magnitude of the total embedded message in terms of bits, let it be *n*.

Step 6. Reinitialize k_i to blank, and for *i* = 1 to $(n-18)/4$ repeat Step 3. Then we get M_i with *n* – 18 bits length.

Step 7. Convert the bits of M_i to characters. The extraction is successful.

4. Results

The given method has been compared with the existing bit flipping method [2], Wu and Tsai [12] and Wu et al. [14] in Tables 1-4. The parameters such as PSNR, hiding capacity, Quality Index (Q.I), and Bit rate (i.e., Bits Per Pixel, BPP), referred from [5, 29] has been considered for comparison. From Table 1 and 3 it may be concluded that the capacity of proposed bit flipping technique is more than that of existing technique. The cover images are shown in Fig. 1(a)-(h) and the resulting images for bit flipping-1, bit flipping-2 schemas respectively are shown in Fig. 2(a)-(h) and Fig. 3(a)-(h).

PSNR tells about the distortion in the resulting image. It is computed by using the Mean Square Error (MSE) among the original image and the resulting image. The larger the PSNR the better technique it is. High PSNR also suggests a lesser distortion in the resulting image. The PSNR with a value more than 40 dB is considered to be a good one. In both the proposed techniques PSNR is acceptable i.e. it is more than 40 dB. The hiding capacity tells about the maximum amount of data bits that can be hidden. The Bit Per Pixel (BPP) parameter tells about the hiding capacity per pixel. A larger BPP value indicates the higher capacity of an image. If we compare among the six techniques which is presented in Tables 1-4 , the average PSNR for eight gray images for Kumar and Chand [2] is 51.27 dB at layer 1 which is superior to other three methods. But at the same time, the average capacity is only 262,144, i.e., one bit per pixel which is very less compared to proposed Bit flipping-1 and 2, Wu and Tsai [12] and Wu et al. [14]. The average capacity for Wu et al. [14] is although higher compared to other methods but its PSNR is 36.07 dB only. The capacity of Bit flipping-2 is doubled compared to the existing method. The PSNR of the proposed techniques are acceptable, i.e., more than 40 dB on an average. Again, the results for 1 LSB and 2 LSB substitutions which provides the PSNR, quality and capacity of 47.09 dB, 0.97, 262,144 and 43.73 dB, 0.93, 524,288 respectively. So based on the above results we claim that our proposed method is better in terms of PSNR and hiding capacity.

If P_{ij} is the m×n gray-scale image and Q_{ij} is its resulting image, then MSE and PSNR values can be computed using next equations:

$$\text{(1)} \qquad \text{MSE} = \frac{1}{m \times n} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(p_{ij} - q_{ij}\right),$$

$$\text{(2)} \qquad \text{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{\text{MSE}}.$$

The quality of the resulting images can be evaluated by using the universal image Quality Index (Q.I) [29]. Q.I tells the equality between the original and the resulting image. It is found by:

$$\text{(3)} \qquad Q = \frac{4\,\sigma_{xy} \bar{p} \bar{q}}{\left(\sigma_x{}^2 + \sigma_y{}^2\right) \left[\, (\bar{p})^2 + (\bar{q})^2 \,\right]},$$

$$\text{(4)} \qquad \bar{p} = \frac{1}{m \times n} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(p_{ij}\right),$$

$$\text{(5)} \qquad \overline{q} = \frac{1}{m \times n} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(q_{ij}\right),$$

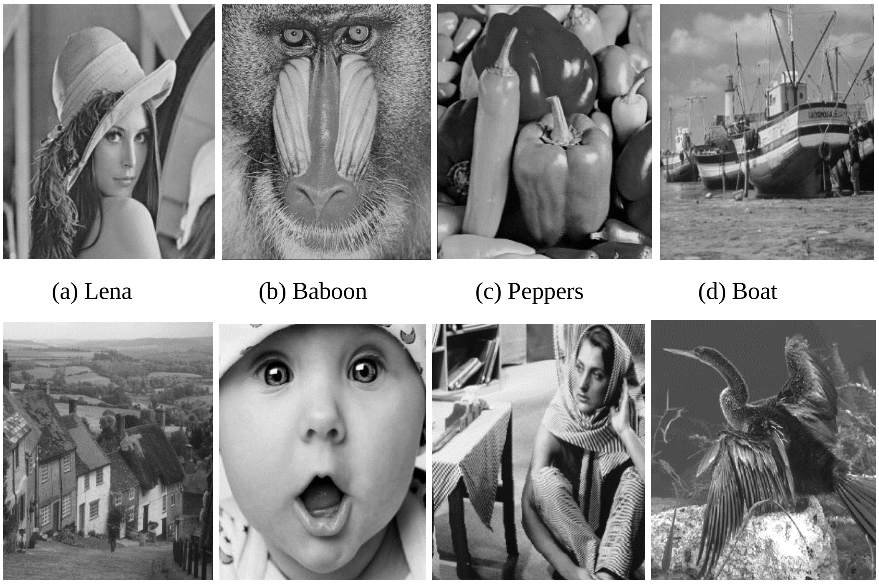
$$\text{(6)} \qquad \sigma_x^2 = \frac{1}{m \times n - 1} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(p_{ij} - \bar{p}\,\right)^2.$$

$$\text{(7)} \qquad \sigma_y^2 = \frac{1}{m \times n - 1} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(q_{ij} - \bar{q}\,\right)^2.$$

$$\text{(8)} \qquad \sigma_{xy} = \frac{1}{m \times n - 1} \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \left(p_{ij} - \bar{p}\right) \left(q_{ij} - \bar{q}\right).$$

5. Security analysis

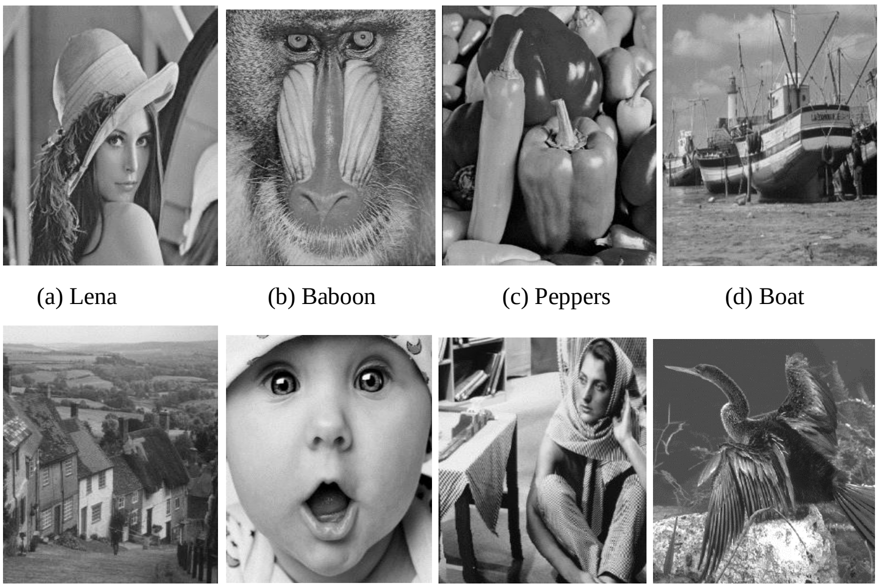
The LSB substitution are exposed to RS steganalytic attacks due to the reason that a pixel value of 2*n* changes to 2*n*+1 but not to 2*n* – 1 similarly the value 2*n*+1 changes to 2*n* but not 2*n*+2, i.e., only two possibilities. But our proposed methods do not depend on only two possibilities. We have 4 different possibilities for Variant-1 and Variant-2 ranging from 00 to 11 for each pixel. Due to this our method escapes from RS steganalytic attacks. So, based upon the above observations we can claim that our method is a secure one.



(a) Lena (b) Baboon (c) Peppers (d) Boat

(e) House (f) Baby (g) Barbara (h) Bird

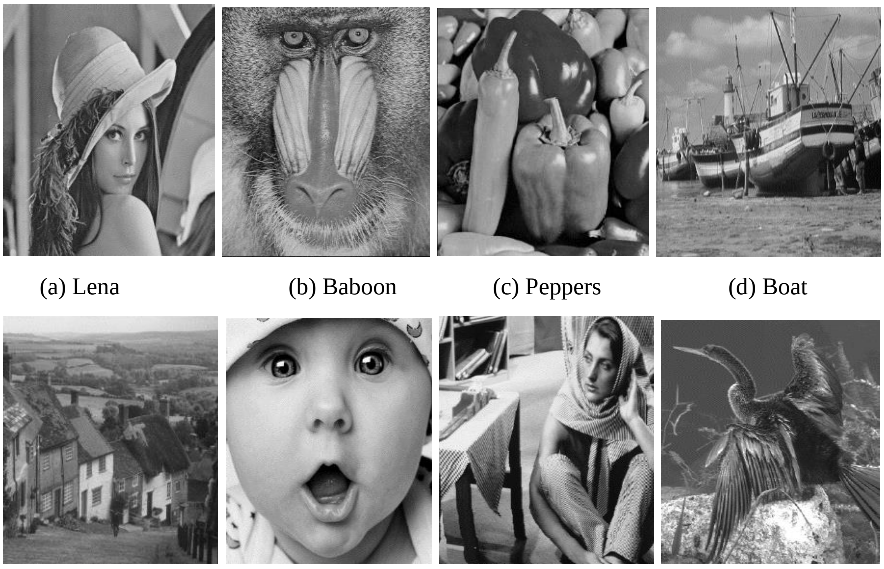
Fig 1. Original images



(a) Lena (b) Baboon (c) Peppers (d) Boat

(e) House (f) Baby (g) Barbara (h) Bird

Fig 2. Resulting images of bit flipping-1



(a) Lena (b) Baboon (c) Peppers (d) Boat

(e) House (f) Baby (g) Barbara (h) Bird

Fig. 3. Resulting images of bit flipping-2

Table 1. Results of existing bit flipping at layer-1 and proposed bit flipping-1 scheme

Images 512×512	Kumar and Chand [2]				Proposed Bit flipping-1			
	PSNR	Capacity	Q.I	BPP	PSNR	Capacity	Q.I	BPP
Lena	51.27	262,144	0.98	1	47.38	393,216	0.97	1.5
Baboon	51.27	262,144	0.99	1	47.36	393,216	0.99	1.5
Peppers	51.28	262,144	0.98	1	47.39	393,216	0.97	1.5
Boat	51.27	262,144	0.99	1	47.20	393,216	0.98	1.5
House	51.28	262,144	0.99	1	47.37	393,216	0.98	1.5
Baby	51.27	262,144	0.97	1	47.36	393,216	0.94	1.5
Barbara	51.27	262,144	0.98	1	47.34	393,216	0.96	1.5
Bird	51.26	262,144	0.89	1	47.80	393,216	0.87	1.5
Average	51.27	262,144	0.97	1	47.40	393,216	0.96	1.5

Table 2. Results of 1-bit and 2-bit LSB substitution

Images 512×512	1-bit LSB substitution				2-bit LSB substitution			
	PSNR	Capacity	Q.I	BPP	PSNR	Capacity	Q.I	BPP
Lena	48.27	262,144	0.98	1	43.58	524,288	0.96	2
Baboon	47.33	262,144	0.99	1	42.76	524,288	0.96	2
Peppers	46.34	262,144	0.99	1	44.39	524,288	0.95	2
Boat	47.78	262,144	0.98	1	43.20	524,288	0.95	2
House	47.71	262,144	0.99	1	43.47	524,288	0.96	2
Baby	47.35	262,144	0.97	1	44.36	524,288	0.92	2
Barbara	45.55	262,144	0.96	1	43.34	524,288	0.92	2
Bird	46.45	262,144	0.91	1	44.80	524,288	0.90	2
Average	47.09	262,144	0.97	1	43.73	524,288	0.94	2

Table 3. Results of Wu and Tsai [12] and Wu et al. [14] scheme

Images 512×512	Wu and Tsai [12]				Wu et al. [14]			
	PSNR	Capacity	Q.I	BPP	PSNR	Capacity	Q.I	BPP
Lena	40.87	409,811	0.97	1.6	37.07	743,616	0.97	2.8
Baboon	36.72	456,993	0.98	1.7	35.16	717,918	0.96	2.7
Peppers	40.71	405,617	0.97	1.5	36.94	768,721	0.97	2.9
Boat	38.87	420,479	0.98	1.6	36.09	756,082	0.97	2.8
House	40.88	411,891	0.98	1.6	37.03	761,618	0.98	2.9
Baby	42.79	395,485	0.96	1.5	37.65	780,407	0.98	2.9
Barbara	33.40	465,654	0.96	1.8	32.60	739,284	0.95	2.8
Bird	38.30	419,615	0.96	1.6	36.05	743,616	0.97	2.8
Average	39.60	423,193	0.97	1.6	36.07	751,407	0.96	2.8

Table 4. Results of Proposed Bit flipping-2

Images 512×512	Proposed Bit flipping-2			
	PSNR	Capacity	Q.I	BPP
Lena	47.31	524,288	0.97	2.0
Baboon	47.33	524,288	0.99	2.0
Peppers	47.32	524,288	0.97	2.0
Boat	47.19	524,288	0.98	2.0
House	47.30	524,288	0.98	2.0
Baby	47.31	524,288	0.94	2.0
Barbara	47.30	524,288	0.96	2.0
Bird	47.66	524,288	0.88	2.0
Average	47.34	524,288	0.96	2.0

6. Conclusion

This article proposes a modified bit flipping technique for hiding information in an image called as bit flipping-1 and bit flipping-2.The proposed bit flipping-1 offers a capacity of 3 bits for a pair of pixels. The bit flipping-2 offers a capacity of 4 bits for a pair of pixels. Thus the capacities of the proposed techniques are improved. Furthermore, it is quite clear from the resulting images that it is not susceptible to the invader.

References

1. Johnson, N. F., S. Jajodia. Exploring Steganography: Seeing the Unseen. – IEEE Computer, Vol. **31**, 1998, No 2, pp. 26-34.
2. Kumar, R., S. Chand. A Reversible Data Hiding Scheme Using Bit Flipping Strategy. – Journal of Discrete Mathematical Sciences and Cryptography, Vol. **19**, 2016, No 2, pp. 331-345.
3. Pradhan, A., A. K. Sahu, G. Swain, K. R. Sekhar. Performance Evaluation Parameters of Image Steganography Techniques. – In: Proc. of International Conference on Research Advances in Integrated Navigation Systems, 2016.
4. Swain, G., S. K. Lenka. Classification of Image Steganography Techniques in Spatial Domain: A Study. – International Journal of Computer Science & Engineering Technology, Vol. **5**, 2014, No 3, pp. 219-232.

5. Wang, Z., A. Bovik. A Universal Image Quality Index. – IEEE Signal Processing Letters, Vol. **9**, 2002, No 3, pp. 81-84.
6. Sahu, A. K., M. Sahu. Digital Image Steganography Techniques in Spatial Domain: A Study. – International Journal of Pharmacy & Technology, Vol. **8**, 2016, Issue 4, pp. 5205-5217.
7. Wang, R., C. Lin, J. Lin. Hiding Data in Images by Optimal Moderately-Significant-Bit Replacement. – Electronics Letters, Vol. **36**, 2000, No 25, pp. 2069-2070.
8. Wang, R. Z., C. F. Lin, J. C. Lin. Image Hiding by Optimal LSB Substitution and Genetic Algorithm. – Pattern Recognition, Vol. **34**, 2001, No 3, pp. 671-683.
9. Chan, C. K., L. M. Cheng. Hiding Data in Images by Simple LSB Substitution. – Pattern Recognition, Vol. **37**, 2004, No 3, pp. 469-474.
10. Swain, G., S. K. Lenka. A Technique for Secret Communication Using a New Block Cipher with Dynamic Steganography. – International Journal of Security and Its Applications, Vol. **6**, 2012, No 4, pp. 13-24.
11. Fridrich, J., M. Goljan, R. Du. Reliable Detection of LSB Steganography in Grayscale and Color Images. – ACM Workshop on Multimedia and Security, 2001, pp. 27-30. **http://doi.org/10.1145/1232454.1232466**
12. Wu, D. C., W. H. Tsai. A Steganographic Method for Images by Pixel-Value Differencing. – Pattern Recognition Letters, Vol. **24**, 2003, No 9-10, pp. 1613-1626.
13. Yang, C. H., C. Y. Weng, S. J. Wang., H. M. Sun. Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems. – IEEE Transactions on Information Forensics and Security, Vol. **3**, 2008, No 3, pp. 488-497.
14. Wu, H. C., N. I. Wu, C. S. Tsai, M. S. Hwang. An Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Methods. – IEE Proceedings on Vision Image Signal Processing, Vol. **152**, 2005, No 5, pp. 611-615.
15. Yang, C. H., C. Y. Weng, S. J. Wang., H. M. Sun. Varied PVD + LSB Evading Detection Programs to Spatial Domain in Data Embedding Systems. – Journal of Systems and Software, Vol. **83**, 2010, No 10, pp. 1635-1643.
16. Wang, C. M., N. I. Wu, C. S. Tsai, M. S. Hwang. A High Quality Steganographic Method with Pixel-Value Differencing and Modulus Function. – Journal of System and Software, Vol. **81**, 2008, pp. 150-158.
17. Liao, X., Q. Y. Wen, J. Zhang. A Steganographic Method for Digital Images with Four-Pixel Differencing and Modified LSB Substitution. – Journal of Visual Communication and Image Representation, Vol. **22**, 2011, No 1, pp. 1-8.
18. Swain, G. Digital Image Steganography Using Nine-Pixel Differencing and Modified LSB Substitution. – Indian Journal of Science and Technology, Vol. **7**, 2014, No 9, pp. 1444-1450.
19. Malik, A., G. Sikka, H. K. Verma. A Modified Pixel-Value Differencing Image Steganographic Scheme with Least Significant Bit Substitution Method. – International Journal of Image, Graphics and Signal Processing, Vol. **4**, 2015, pp. 68-74.
20. Pradhan, A., K. R. Sekhar, G. Swain. Digital Image Steganography Based on Seven Way Pixel Value Differencing. – Indian Journal of Science &Technology, Vol. **9**, 2016, No 37, pp. 1-11.
21. Pradhan, A., K. R. Sekhar, G. Swain. Digital Image Steganography Combining LSB Substitution with Five Way PVD in 2×3 Pixel Blocks. – International Journal of Pharmacy and Technology, Vol. **8**, 2016, No 4, pp. 22051-22061.
22. Swain, G. A Steganographic Method Combining LSB Substitution and PVD in a Block. – Procedia Computer Science, Vol. **85**, 2016, pp. 39-44.
23. Swain, G. Adaptive Pixel Value Differencing Steganography Using Both Vertical and Horizontal Edges. – Multimedia Tools and Applications, Vol. **75**, 2016, No 21, pp.13541-13556.
24. Swain, G. Digital Image Steganography Using Variable Length Group of Bits Substitution. – Procedia Computer Science, Vol. **85**, 2016, pp. 31-38.
25. Juneja, M., P. S. Sandhu. Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption. – In: Proc. of International Conference on Advances in Recent Technologies in Communication and Computing, 2009, pp. 302-305.
26. Swain, G., S. K. Lenka. LSB Array Based Image Steganography Technique by Exploring the Four Least Significant Bits. – In: Proc. of Global Trends in Information Systems and Software Applications, CCIS, 2012, pp. 479-488.

27. Swain, G., S. K. Lenka. A Novel Steganography Technique by Mapping Words with LSB Array. – International Journal of Signal and Imaging Systems Engineering, Vol. **8**, 2015, No 1, pp. 115-122.

28. Sahu, A. K., G. Swain. A Review on LSB Substitution and PVD Based Image Steganography Techniques. – Indonesian Journal of Electrical Engineering and Computer Science, Vol. **2**, 2016, No 3, pp. 712-719.

29. Sahu, A. K., G. Swain. An Improved Information Hiding Method Using Group of Bits Substitution. – International Journal on Communications Antenna and Propagation (IRECAP), Vol. **7**, 2017, No 2, pp. 162-167.

Received 05.07.2017; *Second Version 28.11.2017*; *Accepted 12.12.2017*