Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 85 (2016) 425 – 432

Procedia Computer Science

International Conference on Computational Modeling and Security (CMS 2016)

LSB Layer Independent Robust Steganography using Binary Addition

Biswajita Datta[a]*, Upasana Mukherjee[b], Samir Kumar Bandyopadhyay[a]

[a]*Dept. of Computer Sc. & Engg., University of Calcutta, Kolkata, India*
[b]*Dept. of Computer Sc. & Engg., St. Thomas' College of Engg. & Tech., Kolkata, India*

Abstract

In the field of security different steganography techniques hides data within the cover media in such a way so that human perception cannot follow it. All of these also try to follow three challenges of steganography i.e. robustness, imperceptibility and capacity. This proposed technique meets these three challenges very efficiently. Here the secret data are not directly embedded within the cover file but the intensity of cover pixel are adjusted in such a way so that at the receiver side the actual target bits are extracted from stego image by performing binary addition. The embedding also performs binary addition among desired number of bits selected from LSB and the two LSBs of the result of binary addition are considered as the interpretation of two target data bits. The maximum change in the intensity value is nominal and is not depends on the number of LSB layer chosen for binary addition. Since the actual data are not hidden thus intruders cannot get it by just using the concept of standard LSB extraction technique. Even though they are able to know the binary addition technique used here then also don't get the actual target bits without knowing the number of LSB layers involved for binary addition. Two data bits are embedded in each pixel so from capacity point of view this technique is two times better than standard LSB technique for steganography.

Peer-review under responsibility of the Organizing Committee of CMS 2016

Keywords: Image Steganography, Binary Addition, Robustness, LSB Layer, Binary Search, Robustness, Imperceptibility.

1. Introduction

Today, Information Security, the practice of defending information from unauthorized access, use, modification, recording or destruction, becomes an important security issue with the rise of Internet. Cryptography is a method

* Corresponding author. Tel.: +91-9330806638
E-mail address:biswa.jita@gmail.com

Peer-review under responsibility of the Organizing Committee of CMS 2016
doi:10.1016/j.procs.2016.05.188

used for encrypting messages to maintain the secrecy of a communication procedure for a long decade [1]. But besides keeping the message secret, it is often necessary to keep the very existence of the message under wraps. Steganography, a new technique for security wraps secret data into a carrier file in such a stealthy way which avoids the arousing of an eavesdropper's suspicion. Now this data hiding technique has been proposed as one of the promising techniques for the purposes of authentication, fingerprinting, security, data mining, and copyright protection. It is originated from Greek words *Steganós* (Covered), and *Graptos* (Writing) which literally means "cover writing", provides data security by hiding the very existence of the secret information [2].

Steganography meets three different challenges: Imperceptibility, Robustness, and Capacity. Imperceptibility refers hiding data in such a way so that it cannot deviate the perceptibility of cover media. Robustness of the secret data refers to preventing eavesdroppers from recovering the secret data until and unless they can able to sense the very existence of it [3]. Capacity, the third one means how much data can be embedded without hampering imperceptibility of cover media. Although these three are much related to each other but they should meet within the steganography without disturbing the other [4]. In our proposed work we try to meet imperceptibility by independently choosing the LSB layer as well as increase the capacity of stego media. At the same time the robustness of the secret data is met by not embedding the actual data within the cover media directly. introduce the paper, and put a nomenclature if necessary, in a box with the same font size as the rest of the paper. The paragraphs continue from here and are only separated by headings, subheadings, images and formulae. The section headings are arranged by numbers, bold and 10 pt. Here follows further instructions for authors.

2. Literature Survey

All Least Significant Bit (LSB) steganography is popular and simple approach to embed information within an image [5]. It is more imperceptible technique but the capacity of stego media is very poor as well as robustness is very low. To improve robustness as well as capacity P. Mahimah et al. in their work consider three channel of cover image for hiding data where one channel is considered as indicator channel. They divide the image in 4 sub images then use either default (i.e. red color) or user defined pixel indicator channel in zigzag manner [6]. To increase robustness B. Karthikeyan et al. embed secret data at two LSBs of random position pixel instead of consecutive pixels [7]. To provide more security to this technique M. Bashardoost et al. in their work first encrypts the secret data, then compressed using LZW compression technique and then embed using The Knight Tour algorithm [8]. Himakshi et al. introduce bi- directional Pixel Value Differencing method for RGB color image where difference between two pixels found in both the side to increase both payload capacity and security [9]. V. Agham provides two level of security by using two keys one for encryption of secret data and another for data hiding. Here four bit positions (5, 6, 7, and 8) of randomly selected pixel are used for embedding secret data [10].

The robustness of steganography techniques are increased either by considering higher LSB Layer or without hiding the actual data or by hiding the data in a robust region. N. Hamid et al. find the most robust regions within the image for hiding data using Speeded-Up Robust Features (SURF) [11]. We in our work increase robustness by not hiding actual data within the cover. Here in this work modulo operator is used for getting the actual data at the receiver side [12]. W. Luo et al. in their work use simple LSB embedding scheme but choose a data hiding region based on an edge adaptive algorithm according to the size of secret message and the difference between two consecutive pixels in the cover image [13]. D. Rawat and V. Bhandari in their work embed MSB secret image in the LSB of red, next two MSB in the two LSB of green plane and next three in the 3 LSB of blue based on the research made by Hecht where it is shown that 65% of all human cones are sensitive to red, 33% to green and 2% to blue [14].

3. Proposed Method

Steganography is the process of inserting a secret message within an image in such a way so that no intruder can even feel its existence. Here our objective is to hide a secret text document within a grey scale image in such a way so that actual target bits are not embedded within the stego image for increasing robustness. For doing this the cover pixel values are adjusted in such a way so that at the receiver side the result of binary addition gives the target secret bits.

3.1. Embedding of target data

The embedding of target data is done on two stages - first the adjusted values are placed in their particular bucket then binary addition is used for hiding data within the cover pixels. Each character of the secret text that is to be embedded is converted to its 7 bit ASCII before start processing. Then we decide the value of n which is the number of bits from LSB of cover pixel involved for binary addition.

Bucketing of adjusted values: In this proposed work n (n>2) bits from LSB of cover pixels are considered for binary addition. If the two least significant bits of this addition result are considered then the four combinations are 00, 01, 10 and 11.The higher order bits of binary addition are ignored here. These combinations have a relation with the number of 1 present within the considerable n bits during binary addition. Table 1 show this relationship between the binary addition and the number of 1 involves within this addition.

Table 1. Relationship between the binary addition and the number of 1 involves within this addition.

No. of 1s within the considerable n (n > 2) bit Binary	Two LSBs of binary addition result	No. of 1s within the considerable n (n > 2) bit Binary	Two LSBs of binary addition result
0	0 0	5	0 1 (101)*
1	0 1	6	1 0 (110)*
2	1 0	7	1 1 (111)*
3	1 1	8	0 0 (1000)*
4	0 0 (100)*		

(* *consider only the highlighted bits*)

If the number of bits of cover pixel involved for binary addition is n so 0 to 2^n- 1 different combination are possible in this case. Now these decimal values are con-verted in binary and then arranged in their specific bucket by counting number of 1s in this binary string. The buckets are prepared according to two LSBs of binary addition result and are named according to the two bit combinations of binary. Table 2 shows the bucketing result for n=4 and n=6.

Table 2. Four buckets by considering 4 and 6 LSBs for addition

Bucket Class	4 Bits from LSB are considered for binary addition	6 Bits from LSB are considered for binary addition
	Numbers 0 to 15	Numbers 0 to 63
00	0, 15	0, 15, 23, 27, 29, 30, 39, 43, 45, 46, 51, 53, 54, 57, 58, 60
01	1, 2, 4, 8	1, 2, 4, 8, 16, 32, 63
10	3, 5, 6, 9, 10, 12	3, 5, 6, 9, 10, 12, 17, 18, 20, 24, 31, 33, 34, 36, 40, 47, 48, 55, 59, 61, 62
11	7, 11, 13, 14	7, 11, 13, 14, 19, 21, 22, 25, 26, 28, 35, 37, 38, 41, 42, 44,49, 50, 52, 56

Embedding using Binary Addition: In this proposed work the embedding technique does not embed the target data directly into the pixel but the desired bits in cover pixels are modified in such a way so that at the receiver side binary addition results of these bits gives the actual data bits.

First the target cover pixels are converted to their binary equivalent of 8 bits, then the binary addition of n (n > 2) desired number of bits from LSB are calculated. After that two target bits are compared with the two least significant bits of addition result, then the desired n bits are adjusted if required.

If two target bits are same with the 2 LSBs of binary addition result then there is no change is made. If these are not same then the adjustment is required. For doing this adjustment first the particular bucket are chosen based on the two target bits. Then the closest value from this bucket is selected to modify n considerable LSBs for addition so that in receiver side result of binary addition of this n number of desired bits gives the actual target bits. Since each of the buckets is arranged in ascending order the binary search technique is applied for choosing the closest value.

Let “01001110…” this target bit string is to be inserted and n = 4 LSBs of cover pixels are considered for binary addition. Suppose two target bits are embedded within a cover pixel of intensity value 189_{10} (10111101_2). The 4 bit from LSB of its 8 bit representation is 1101_2 (13_{10}). Now binary addition is performed among these 4 bits (1 + 1 + 0 +

1 = 11) and the last two bits of this addition result (i.e. 11_2) is compared with the two target bits (01_2). These are not similar. Thus the 4 bits (1101_2 or 13_{10}) is replaced with the closest value of '01' bucket and here it is 1000_2 (8_{10}). Now 189_{10} (10111101_2) is replaced with 184_{10} (10111000_2).

In this proposed technique since 7 bit ASCII of the character is considered and 2 bits of target text are embedded at a time. So the size of the target bit stream should be even, if it is not then make it even by padding 0 at the end. The target string length and the desired number of bits used for binary addition are inserted at the beginning of the stego image by using same binary addition concept.

3.2. Extraction of target data

At the receiver side first the string length and the numbers of bits (n > 2) involved for binary addition are extracted then the target hidden data are extracted using binary addition. During each iteration, first the pixel intensities are converted to their corresponding 8 bit binary. Then the binary addition is performed among the n (n > 2) LSBs and the two least significant bits of summation result are obtained. Let, extracted value of n is 4 and a stego pixel value is 184_{10} (10111000_2). The 4 bit from LSB of 10111000_2 is 1000_2. Now the last two bit of binary addition result of 1000_2 is 01 (1 + 0 + 0 + 0 = 01). This 01 is the target secret bit extracted at the receiver side.

This entire procedure continues until the recipient extracts the entire target bit stream sent by the sender and store these results. Now these pair of bits are concatenated to make the target bit stream. Then each of the 7 bits are cut and converted to their corresponding ASCII. The corresponding target text is generated from these ASCII.

4. Algorithm

4.1. Algorithm for Embedding

Step 1: Take an image and a text as input. Convert the text into binary.
Step 2: Convert the text into binary.
Step 3: Store the target string length as well as the value of n within the cover image.
Step 4: Place the values within the range 0 to 2^n-1 into its particular bucket.
Step 5: Consider the last n bit of cover pixels of the image and perform binary addition.
Step 6: Now take the pair of bits of the message and compare it with the two LSBs of addition result.
Step 7: If the binary addition value is same with the last two message bits then nothing is needed to be done, otherwise the n bits will be adjusted by the closest value of particular bucket.
Step 8: Repeat step 4 to step 6 for embedding all target bits.
Step 9: Send the stego message to the receiver side.
Step 10: End

4.2. Algorithm for Extraction

Step 1: Take the stego image as input.
Step 2: Consider the last n bits (confirm by sender) of each target pixel of the stego image and perform binary addition.
Step 3: Two LSBs of addition result are considered as the bits of secret data putting from the end.
Step 4: Now concatenate 7 bits each and convert them to their corresponding decimal value.
Step5: Write it into another file as a form of characters and form the target message.
Step 6: End

5. Result Analysis

This section first shows the simulated result of the proposed technique. Then analyze these results based on PSNR, NCC, Average Difference (AD) and LMSE calculation, and also by capacity of cover media. Also a standard benchmark tool – StirMark Benchmark is used for performance analysis of proposed method. The testing is done on

500 and more images of different categories such as scenery, cartoon, jungles, single object, crowd, motion, text, map etc. Analysis is done based on embedding two types of text messages. First one affects most of the cover pixels (Large Text) and second one covers half of the pixels of cover images (medium Text). Some sample results from the testing database are shown in Fig. 1.

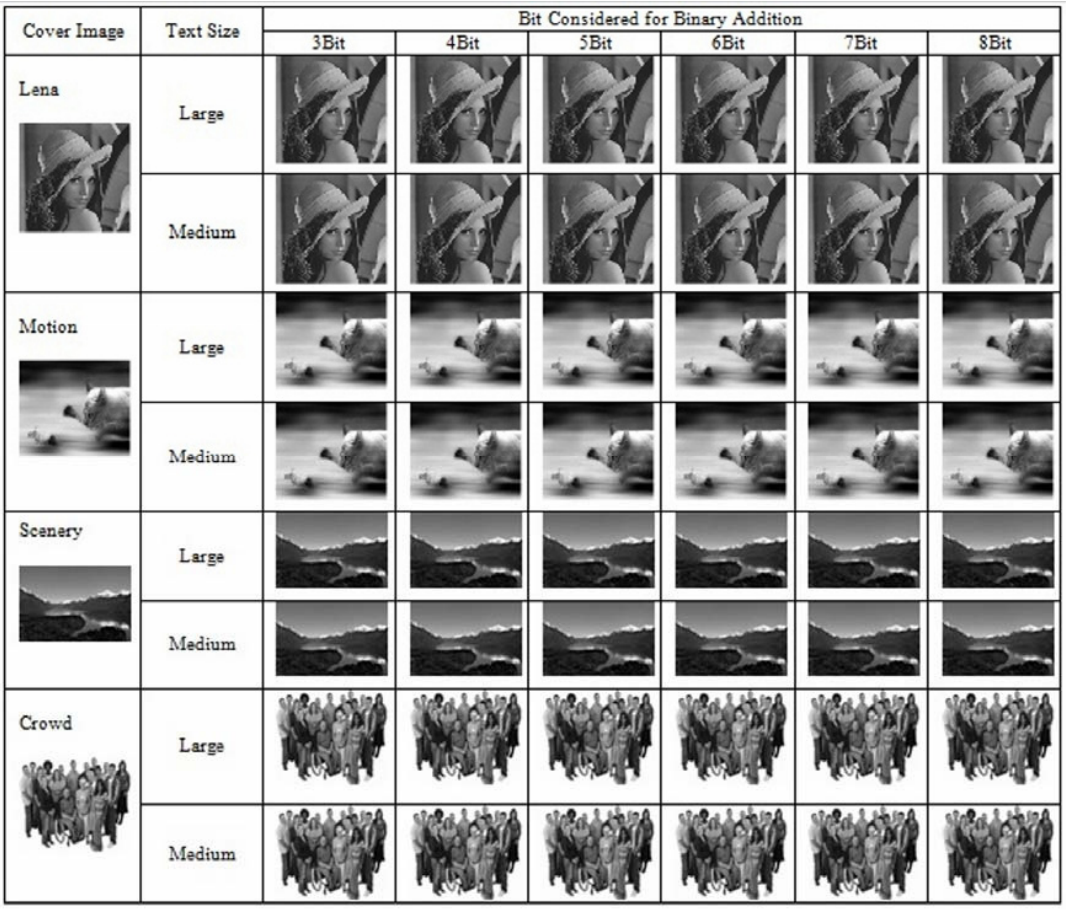


Fig. 1. Sample Stego images after embedding large and medium size text by considering 3, 4, 5, 6, 7 and 8 bits from LSBs of cover images.

Now the image quality of the output stego images is analysed by MSE, PSNR, LMSE, NCC and AD.

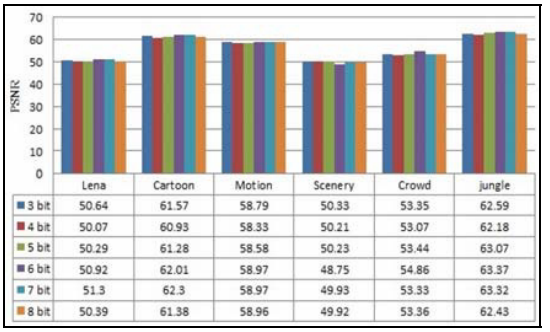
The *Peak Signal to Noise Ratio* (PSNR) is used to measure the quality of stego image compared to the cover image. Higher PSNR indicates that the reconstruction of the image is of higher quality. It is calculated as,

$$PSNR = 10 \times \log\left(\left(\frac{255^2}{MSE}\right)\right) \quad (1) \qquad \text{where, } MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|c-s\|^2 \quad (2)$$

Where m, n are the width and height of the image and c is the cover image and s is the stego image. In Fig. 2 and Fig. 3 PSNR values of sample images are shown after embedding large and medium sized text respectively.

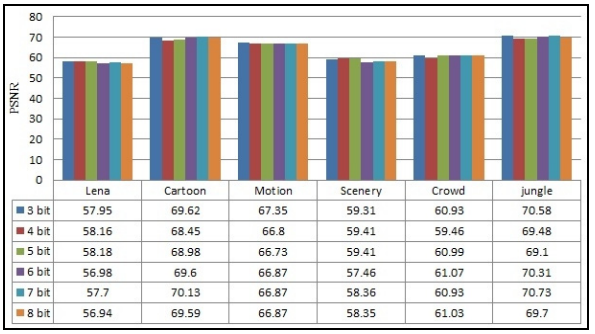
The PSNR result is almost same in the case of proposed technique either by choosing 3, 4, 5, 6, 7 or 8 LSB bits for binary addition because in all these cases the maximum changes is 7 after adjusting the bits for embedding.

The large value of *Laplacian Mean Square Error* (LMSE) means that image is poor quality. LMSE should be 0 for identical image. If there is any error, LMSE value may between 0-1.It is calculated as,



	Lena	Cartoon	Motion	Scenery	Crowd	jungle
3 bit	50.64	61.57	58.79	50.33	53.35	62.59
4 bit	50.07	60.93	58.33	50.21	53.07	62.18
5 bit	50.29	61.28	58.58	50.23	53.44	63.07
6 bit	50.92	62.01	58.97	48.75	54.86	63.37
7 bit	51.3	62.3	58.97	49.93	53.33	63.32
8 bit	50.39	61.38	58.96	49.92	53.36	62.43

Fig. 2. PSNR for Large sized text embedding



	Lena	Cartoon	Motion	Scenery	Crowd	jungle
3 bit	57.95	69.62	67.35	59.31	60.93	70.58
4 bit	58.16	68.45	66.8	59.41	59.46	69.48
5 bit	58.18	68.98	66.73	59.41	60.99	69.1
6 bit	56.98	69.6	66.87	57.46	61.07	70.31
7 bit	57.7	70.13	66.87	58.36	60.93	70.73
8 bit	56.94	69.59	66.87	58.35	61.03	69.7

Fig. 3. PSNR for Medium sized text embedding

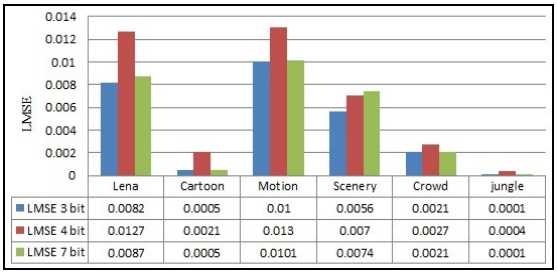
$$LMSE = \frac{\sum_{0}^{m-1}\sum_{0}^{n-1} OP - 4\times\nabla^{2}(s)^{2}}{\sum_{0}^{m-1}\sum_{0}^{n-1} OP^{2}} \quad (3) \qquad \text{where, } OP = 4\times\nabla^{2}(c) \quad (4)$$

Normalized Cross Correlation (NCC) is a measure of similarity of two waveforms as a function of the time lag applied to one of them i.e. convolution of two functions. NCC should be 1 for identical image. It is calculated as,

$$NCC = \sum_{0}^{m-1}\sum_{0}^{n-1}(c\times s)/c^{2} \quad (5)$$

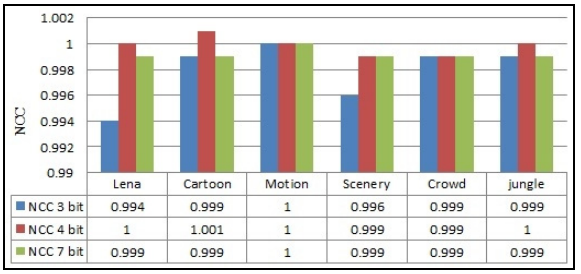
where c and s are cover and stego images.

LMSE, NCC analysis results are shown in Fig. 4 and Fig. 5 based on 3, 4 and 7 bit binary addition on some set of sample images and by considering large sized text.



	Lena	Cartoon	Motion	Scenery	Crowd	jungle
LMSE 3 bit	0.0082	0.0005	0.01	0.0056	0.0021	0.0001
LMSE 4 bit	0.0127	0.0021	0.013	0.007	0.0027	0.0004
LMSE 7 bit	0.0087	0.0005	0.0101	0.0074	0.0021	0.0001

Fig. 4. LMSE for Large sized Text embedding



	Lena	Cartoon	Motion	Scenery	Crowd	jungle
NCC 3 bit	0.994	0.999	1	0.996	0.999	0.999
NCC 4 bit	1	1.001	1	0.999	0.999	1
NCC 7 bit	0.999	0.999	1	0.999	0.999	0.999

Fig. 5. NCC for Large sized Text embedding

The efficiency of our proposed method is also analysed based on StirMark Benchmark Version 4.0. It considers different tools like Embedded/Extraction time, Add noise, Median Cut, Convolution filter, cropping, rotation, PSNR and many more for comparing cover and stego images. In Table 3 some sample benchmarked results are shown by considering 4 LSBs for binary addition.

We analyse capacity of cover media based on the number of target bits embedded within that media. Capacity is again inversely proportional to robustness as well as perceptibility. Our aim is to embed more target data without disturbing robustness as well as imperceptibility. The analysis of proposed work based on capacity is shown in Table 4.

Table 3. Sample result of Stirmark benchmark

Analysed Image	Embed time ms	Add noise 20 dB	Jpeg 100 dB	Median cut 7 dB	Conv. filter dB	Remove lines 60 NA	Cropping 20 NA	Rescale 50/200 NA	Rotation NA -2/2	Affine NA 2/4
lena	5.8	9.114	61.439	25.139	10.328	108.559	119.966	107.87	102.272	103.66
					-3.391			108.64	102.25	103.67
Lena_s	5	9.112	61.163	25.130	10.326	108.598	119.966	107.91	102.293	103.70
					-3.394			108.67	102.274	103.70
jungle	68.6	11.510	59.238	17.312	13.590	67.827	105.858	67.89	63.199	63.54
					3.219			67.75	63.198	65.33
jungle_s	68.4	11.509	59.236	17.312	13.590	67.829	105.858	67.89	63.199	63.54
					3.218			67.75	63.200	65.33
motion	22.2	7.746	62.582	34.435	12.272	136.095	162.807	136.44	126.911	127.04
					-4.523			136.40	126.912	131.97
motion_s	25	7.746	62.480	34.425	12.272	136.102	162.809	136.45	126.914	127.05
					-4.524			136.41	126.915	131.98
cartoon	25.79	7.415	60.675	23.523	11.875	146.376	177.976	146.2	136.731	137.34
					-4.129			146.25	136.731	141.12
cartoon_s	28	7.417	60.675	23.319	11.87	146.373	177.959	146.21	136.733	137.34
					-4.354			146.24	136.732	141.12

Table 4. Capacity analysis

Cover Image Name	Cover Image Size KB	Dimension (8 bit grey image)	No. of Characters in the Text File (c)	No. of bits in the Text File (A = c*7)	No. of Pixels Required (B=A/2)	% of used cover pixel
Jungle	769	1024×768	6784	47488	23744	3.02%
Motion	319	700×466	1168	8176	4088	1.25%
Lena	51.1	225×225	160	1120	560	1.11%

In our proposed technique the bits of the cover images are not replaced by the actual target data bits, only the bits of the cover images are adjusted in such a way so that at the receiver side the result of the binary addition gives the target bits. Here the binary addition technique is also used for embedding two target bits at a time and we consider two least significant bits of summation for the interpretation of target data. Since two target bits are embedded at a time so from capacity point of view these proposed technique is two times better than the standard LSB technique.

The actual target bit is not embedded within the cover pixel here thus if any intruder can able to understand the existence of hidden message then also he cannot get it by using LSB technique concept on any bit position. If that unauthorized person already knows the application of binary addition in this proposed work then also he cannot get the actual data by applying binary addition approach without knowing how many LSBs are involved for this addition. For example if the stego pixel is 10110110_2 and 3, 4, 5, 6, 7, 8 any number of LSBs are considered for binary addition then the results of two least significant bits of summation are not same in all cases which is shown in Table 5. So our proposed technique is efficient from robustness point of view.

The maximum change in pixel intensity is independent of number of considerable bits for embedding which makes our proposed technique more imperceptible. If value of n is greater than 2 i.e. any number of Least Significant bits more than 2 is involved for performing binary addition the maximum change is 7 for all cases. Each time the bucket of four pairs 00, 01, 10 and 11 are form before embedding. During the adjustment always the nearest values from the particular bucket is chosen. If the four buckets form by considering any number of bits (always greater than 2) are examined then we show that the smallest values are 0, 1, 3 and 7 (in decimal) for bucket 00, 01, 10 and 11 respectively.

Table 5. Two LSBs of addition result by considering different number of bits from LSB

No. of bits consider for binary addition	Bits considered for addition	Result of two least significant bits of summation
3	10110110	10
4	10110110	10
5	10110110	11
6	10110110	00
7	10110110	00
8	10110110	01

Four buckets by considering 4 and 6 LSBs for binary addition are shown in Table 2. If the 4 least significant bits are considered then the maximum change (value 7) in pixel intensity value is occurred when say last four bits is 0_{10} (i.e. 0000_2) and the target bit pair is 11_2 or vice versa. In this situation we need to adjust last four bits of cover pixel by the nearest value of 11 bucket i.e. by 7_{10} (i.e. 0111_2) to get 11_2 at the receiver side only by binary addition of 4 LSBs. This discussion is true for any number of bits (should be greater than 2) considered for binary addition.

6. Conclusion

Steganography becomes a most trustworthy security technique in today's communication world. Different techniques are available in this field for providing security and they can try to meet three challenges of steganography by their own. But the technique discussed in this paper meets three challenges of steganography more efficiently. It meets imperceptibility issue by ensuring that the maximum number of changes in intensity value is independent of number of LSB layer chosen for binary addition. It is robust in the sense that the actual target bits are not embedded within the cover pixel to get the stego image. It is two times capacitive than standard LSB technique for steganography. The capacity can also be increased by considering three target bits at a time instead of two. But in this case maximum change in pixel intensity value becomes more (since the number of chosen bits always greater than 6) which affect imperceptibility issue.

References

1. Petitcolas FAP, Anderson R J, Kuhn MG. Information Hiding - A Survey. In Proc. IEEE, vol. 87, no. 7, pp. 1062-1078; July 1999.
2. Provos N, Honeyman P. Hide & Seek: An Introduction to Steganography. IEEE Security & Privacy Magazine, pp 32 -44; 2003.
3. Dunbar B. Steganographic techniques and their use in an Open-Systems environment. SANS Institute; January 2002.
4. Artz D. Digital Steganography: Hiding Data within Data. IEEE Internet Computing Journal; June 2001.
5. Nguyen BC, Yoon SM, Lee HK. Multi Bit Plane Image Steganography. IWDW, LNCS, vol. 4283, pp. 61–70; 2006.
6. Mahimah P, Kurinji R. Zigzag Pixel Indicator based Secret Data Hiding Method. In Proc. of IEEE International Conference on Computational Intelligence and Computing Research; 2013.
7. Karthikeyan B, Vaithiyanathan V, Thamotharan B, Gomathymeenakshi M, Sruti S. LSB Replacement Steganography Using Psudoranomise Key Generation. Research Journal of Applied Sciences, Engineering and Technology; 2012.
8. Bashardoost M, Sulong GB, Parisa G. Enhanced LSB Image Steganography Method By Using Knight Tour Algorithm, Vigenere Encryption and LZW Compression. International Journal of Computer Science; 2013.
9. Himakshi, Verma HK, Singh RK, Singh CK. Bi-Directional pixel-value differencing approach for RGB Color Image. In Proc. of IEEE Sixth International Conference on Contemporary Computing (IC3), pp. 47-52; 2013.
10. Agham V, Pattewar T. A Novel Approach Towards Separable Reversible Data Hiding Technique. In Proc. of IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 771-775; 2014.
11. Hamid N, Yahya A, Ahmad RB, Al-Qershi OM. Characteristic region based image steganography using Speeded-Up Robust Features technique. In Proc. IEEE International Conference on Future Communication Networks (ICFCN 2012), pp. 141-146, Iraq; 2012.
12. Datta B, Bandyopadhyay SK, Kedia A. High Imperceptible Data Hiding using Remainder Method. International Journal of Computer Applications, Vol. 95, No.18, pp. 12-19 ; June 2014.
13. Luo W, Huang F, Huang J. Edge Adaptive Image Steganography Based on LSB Matching Revisited. IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2; June 2010.
14. Rawat D, Bhandari V. A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image. International Journal of Computer Applications, Vol. 64, No. 20; February 2013.