A New Image Steganography Approach for Information Security Using Gray Level Images in Spatial Domain

Rajkumar Yadav*, Ravi Saini and Kamaldeep

University Institute of Engineering & Technology

M.D. University Rohtak. Haryana (India)

rajyadav76@rediffmail.com; ravisaini1988@rediffmail.com;kamalmintwal@gmail.com;

Abstract

A new image steganography method for hiding data using Gray Level Images in Spatial Domain is proposed in this paper. This method uses the 5^{th}, 6^{th} and 7^{th} bits of pixel value for insertion and retrieval of message by using the same bits of pixel value. This method is an improvement over earlier methods like least significant bit (LSB) method [4] and gray level modification (GLM) method [6]. This method retains the advantages of above said methods but discards the disadvantages associated with above methods and provides us the better results.

Keywords:

LSB method, GLM method, cryptography, steganography, pseudo random number generator.

1. Introduction

Steganography is an art and science of hiding information in some cover media. The term originated from Greek roots literally mean "covered writing" [1]. The field of steganography is very old. The most popular stegnographic methods used by spies include invisible ink and microdots. People used etching messages in wooden tablets and covered with wax. They used tatooing a shaved messenger's head, letting his hair graw back and then saving it again when he arrived at his contact point to reveal the message. [2].

Digital Steganography uses the digital objects such as image, music, video or any other computer file for hiding the data. The idea was first given by Simmons in 1983 [5]. Steganography is different from cryptography which is about concealing the content of message whereas steganography is to conceal the very presence of message [7]. The most popular and oldest technique for hiding data in digital image is LSB technique [4]. Another technique for hiding data in digital image is GLM (Gray Level Modification) technique [6]. The advantage of both above methods is that they are simple to implement and invisible to human eye. But there are two disadvantages that are associated with least significant bit (LSB) method as well as gray level modification (GLM) method. First disadvantage of above methods is that if the intruder changes least significant bit (LSB) of all image pixels than hidden message can be destroyed but the change in image quality is in the range of +1 to -1 at each pixel position which is negligible to human eye. Second disadvantage is that least significant bit may be corrupted by hardware imperfections or quantization noise [8] due to which message can be distorted.

In Gray Level Modification method if least significant bit (LSB) changes due to above two problems then pixel value become from even to odd or odd to even due to which message can be destroyed. In our method, we are using 5^{th}, 6^{th} and 7^{th} bit for insertion and retrieval of message. Our method removes both disadvantages associated with LSB and GLM technique and provide us better results.

2. Description of proposed method

In the proposed method, we have used 5^{th}, 6^{th} and 7^{th} bits of pixel value and according to our approach if decimal value of 5^{th} , 6^{th} and 7^{th} bits are 0, 2, 4 or 6 then iinsert 0 at these locations If decimal value of 5^{th}, 6^{th} and 7^{th} bit are not 0, 2, 4 or 6 then we add or subtract 1 at that location for making decimal value of 5^{th}, 6^{th} and 7^{th} bit 0, 2, 4 or 6 for insertion of 0. Similarly, we can insert 1 at a pixel location if decimal value of 5^{th}, 6^{th} and 7^{th} bit at that location is 1, 3, 5 or 7. If decimal value of 5^{th}, 6^{th} and 7^{th} bit at that location is not 1, 3, 5 or 7 then we add or subtract 1 at that location for making decimal value of 5^{th}, 6^{th} and 7^{th} bit 1, 3, 5 or 7 for insertion of 1. The insertion process is shown in figure 1(a).

For retrieval of message, we again check decimal value of 5^{th}, 6^{th} and 7^{th} bit. If the decimal value of 5^{th}, 6^{th} and 7^{th} bit at the selected location is 0, 2, 4 or 6, then 0 is the message bit else message bit is 1. The retrieval process is shown in Figure 1(b).

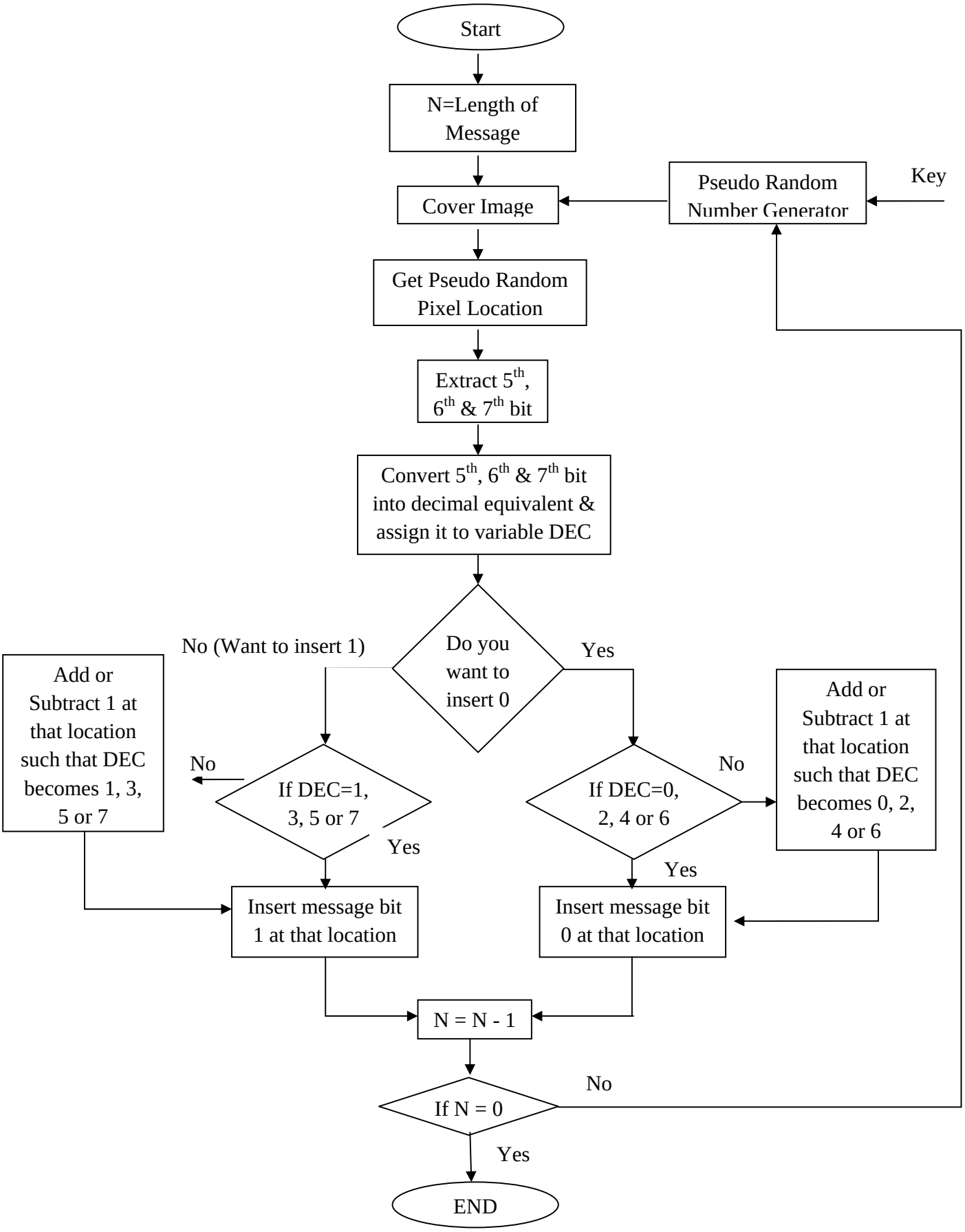

Figure 1 (a) Insertion Process

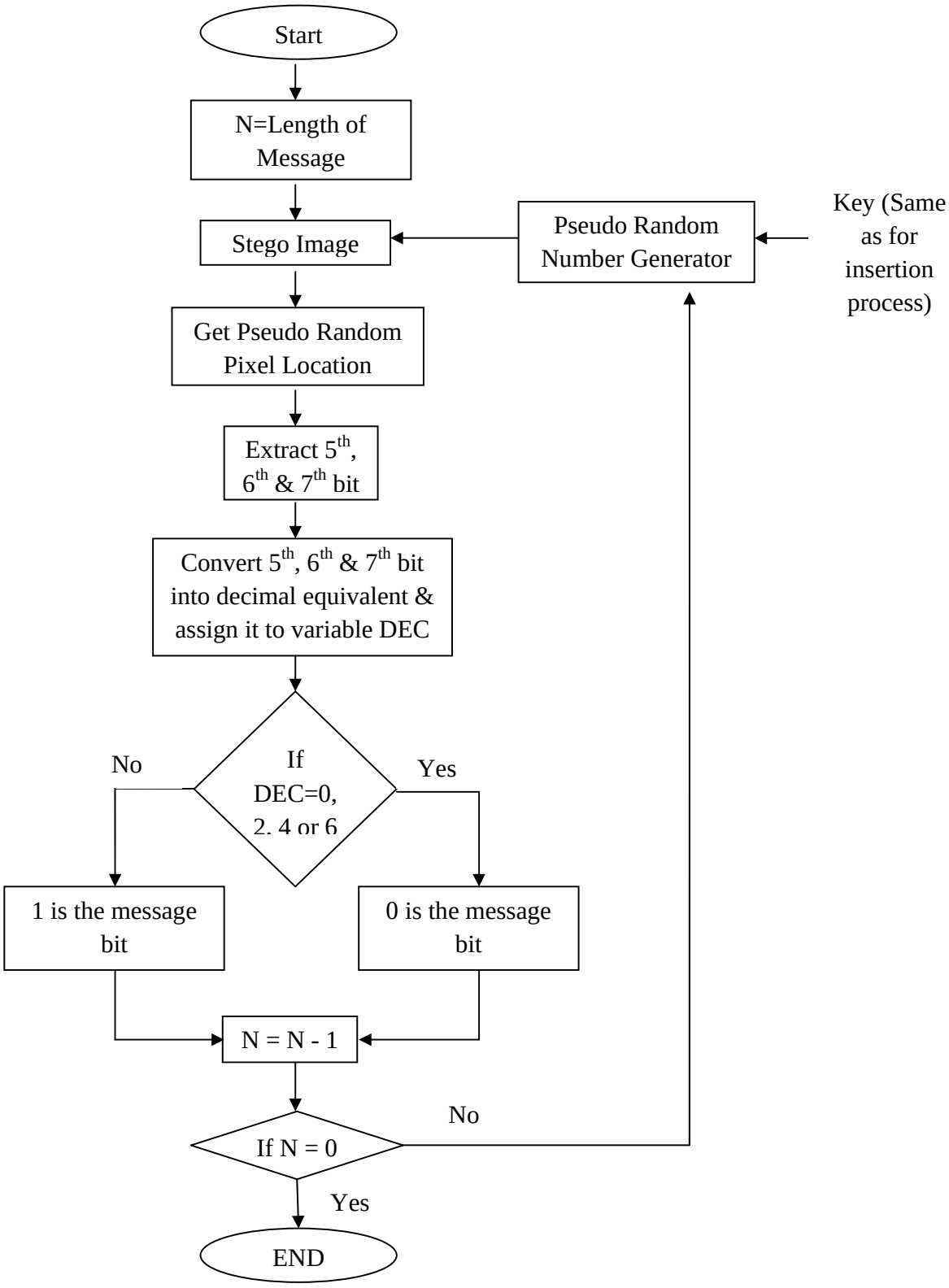

Figure1 (b) Retrieval Process

3. Algorithms

3.1 Assumption

(i) Sender and Recipient agree on the cover image in which message is to be hidden.
(ii) Both sender and recipient agree on the same pseudo-random key to decide the random locations where message is to be inserted.

3.2 Insertion Algorithm

(i) Find pseudo- random location (L) in cover image from secret key to insert the message bit. (For detail see [3] and [9]).
(ii) Check whether at location (L); pixel value is 00000000 or 11111111. If yes, ignore this location and go to step (i). Here, we are ignoring these boundary values because change may be +2 or -2 in pixel values which is to be avoided.
(iii) Extract the 5^{th}, 6^{th} and 7^{th} bit of selected location (L).
(iv) Convert 5^{th}, 6^{th} and 7^{th} bit into equivalent decimal value and assign it to variable DEC.
(v) If we want to insert 0 then go to step (vi) else go to step (vii).
(vi) (a) Check whether DEC = 0, 2, 4 or 6. If yes, insert 0 at location 'L' and go to END.
(b) If DEC is not equal to 0, 2, 4 or 6 then add or subtract 1 at location (L) for making DEC = 0, 2, 4 or 6 at that location and insert 0 at location (L). Go to END.
(vii) (a) Check whether DEC = 1, 3, 5 or 7. If yes, insert 1 at location 'L' and go to END.
(b) If DEC is not equal to 1, 3, 5 or 7 then add or subtract 1 at location (L) for making DEC = 1, 3, 5 or 7 at that location and insert 1 at location (L). Go to END.
(viii) END.

3.3 Retrieval Algorithm

(i) Trace out the location (L) from the same secret key as used for insertion of message.
(ii) Pixel value is 00000000 or 11111111? If yes, then it is invalid location. Go to step (i).
(iii) Extract 5^{th}, 6^{th} and 7^{th} bit from location (L).
(iv) Convert 5^{th}, 6^{th} and 7^{th} bit into equivalent decimal value and assign it to variable DEC.
(v) (a) If DEC = 0, 2, 4 or 6 then 0 is the message bit.
(b) If DEC = 1, 3, 5 or 7 then 1 is the message bit.
(vi) END.

4. Example of proposed method

Suppose a Grayscale image has the following pixels as shown in Figure 2(a) and we have to insert the message 100101 in that image. First we have to insert the message bits. The selection of pixel locations is done by using the pseudo random number generator. Pseudo random number generator uses the same key for insertion as well as retrieval of process.

Columns

Rows	1	2	3	4	5	6	7	8	9	10
1	11	15	20	18	33	79	89	84	210	213
2	217	137	25	85	27	23	45	84	71	215
3	21	45	46	90	112	125	118	17	77	81
4	81	64	65	84	29	210	201	245	139	174
5	11	91	71	206	154	109	219	135	31	87
6	41	47	147	98	87	97	35	135	140	104
7	201	215	115	83	53	43	95	10	47	65
8	95	68	35	64	74	46	32	16	86	65
9	23	16	18	85	43	214	201	105	10	101
10	51	47	29	16	84	47	65	35	38	49

Figure 2 (a) Gray Level values of an Image

Pseudo random number generator generates the random pixel locations by using same key. The location generated by pseudo random number generator is shown in figure 2(b). These locations can be any locations depend upon the key.

Columns →

Rows ↓	1	2	3	4	5	6	7	8	9	10
1	11	15	20	18	33	79	89	84	210	213
2	217	137	25	85	27	23	45	84	71	215
3	21	45	46	90	112	125	118	17	77	81
4	81	64	65	84	29	210	201	245	139	174
5	11	91	71	206	154	109	219	135	31	87
6	41	47	147	98	87	97	35	135	140	104
7	201	215	115	83	53	43	95	10	47	65
8	95	68	35	64	74	46	32	16	86	65
9	23	16	18	85	43	214	201	105	10	101
10	51	47	29	16	84	47	65	35	38	49

Figure2 (b) Pixels selected using Pseudo Random Number Generator

The locations are generated in the order

L (4, 5), L (6, 1), L (10, 9), L (6, 7), L (2, 9), L (8, 4)

where L (i, j) means location at intersection of row number i and column number j.

The corresponding Gray level values regarding these locations are 29, 41, 38, 35, 71 and 64.

The insertion of message 100101 is shown in Figure 2 (c).

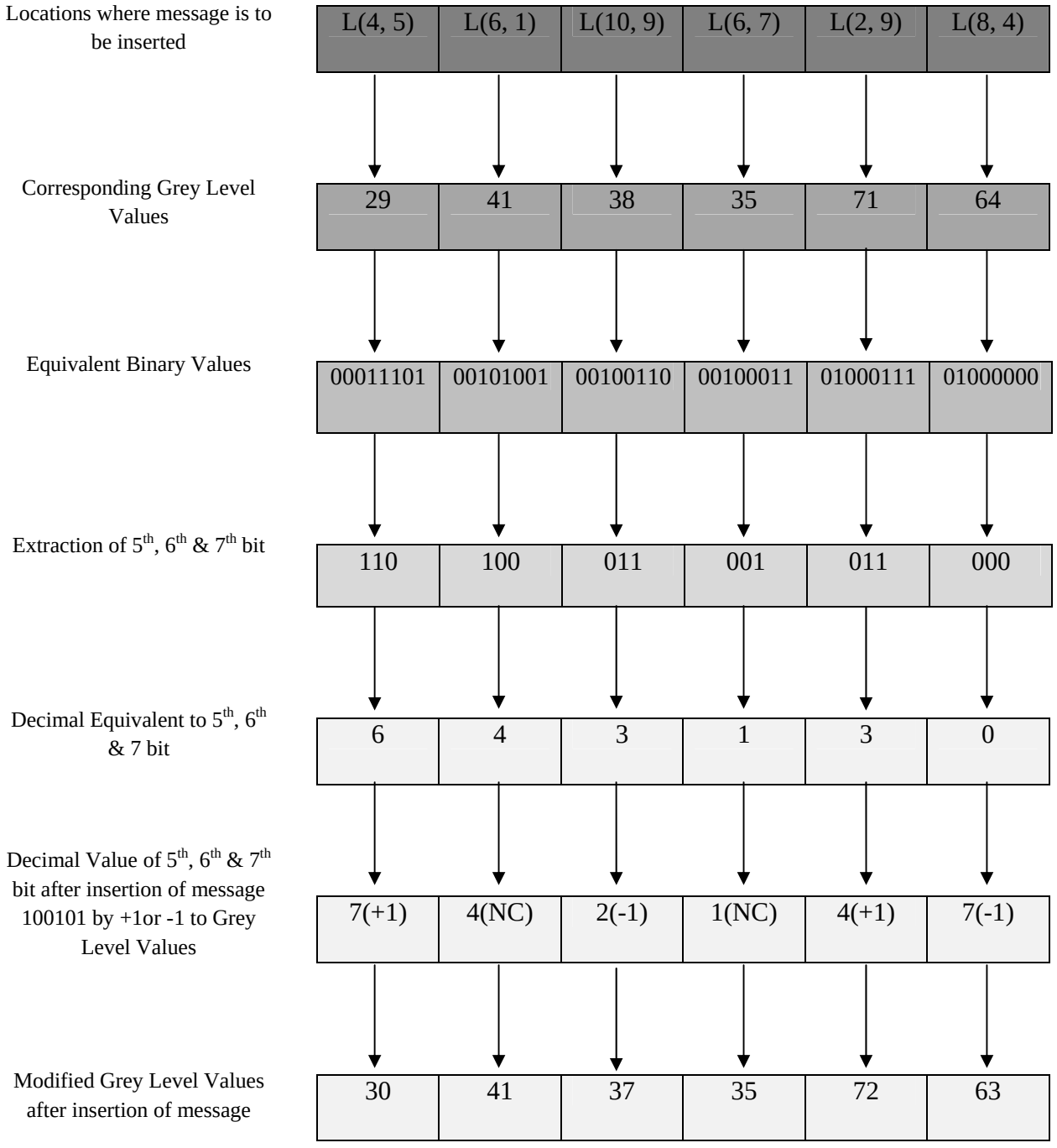

Figure 2 (c) Insertion of message 100101

The retrieval of message from the same locations of cover image is shown in figure 2 (d).

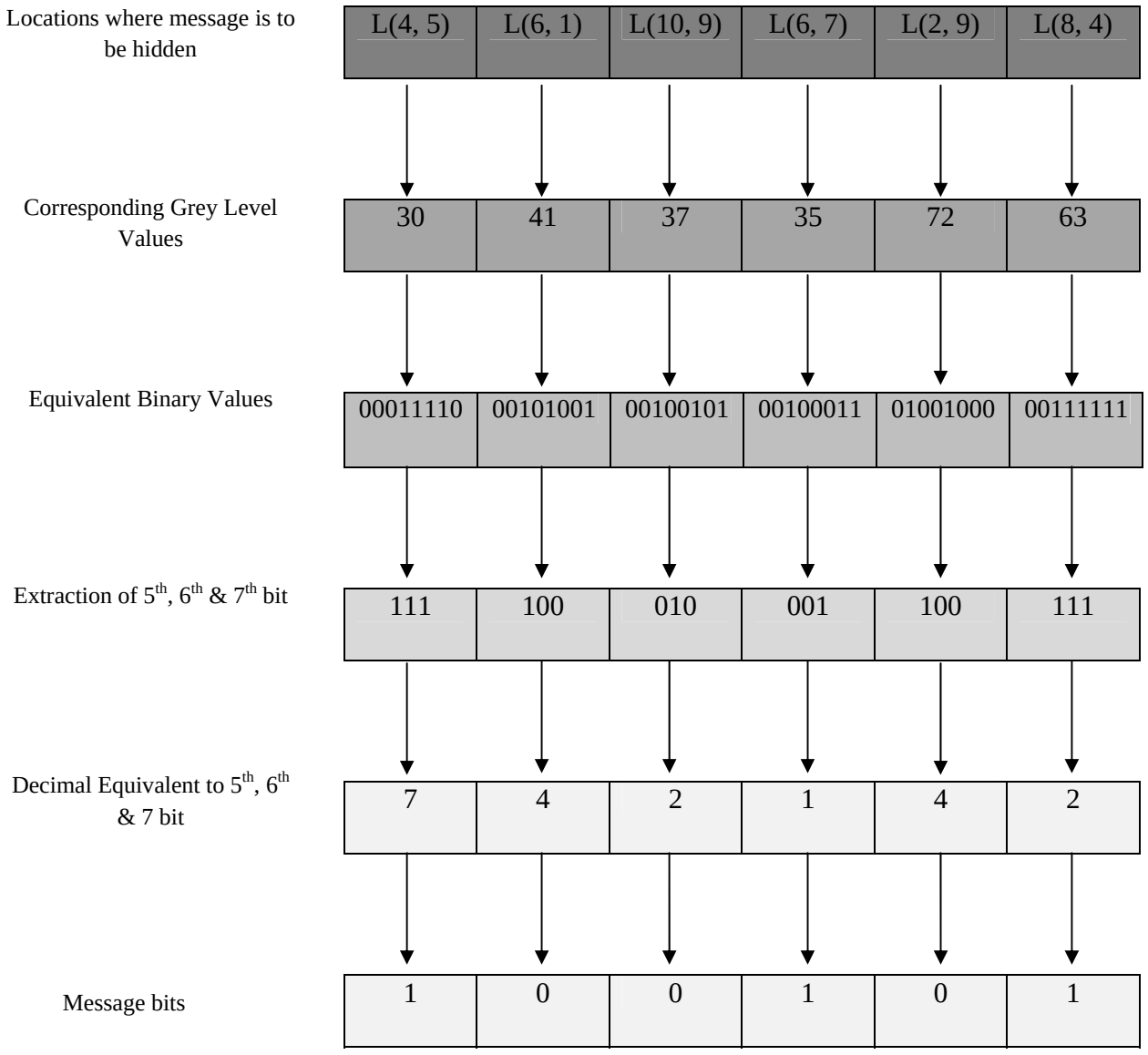

Figure 2(d) Retrieval of message 100101

5. Change in pixel during insertion of message

Now, we see how various pixel values are changed during insertion process. There are two tables shown for insertion of 0 and 1 separately. Table I shows how pixel values changes during insertion of 0 and Table II shows how pixel values changes during insertion of 1.

Table - I

Decimal Value	Pixel value before insertion of '0'	Extraction of 5^{th}, 6^{th} & 7^{th} by bit extractor	Decimal value of 5^{th}, 6^{th} & 7^{th} bit	Pixel value after insertion of '0'	Change in Pixel value & comment for insertion of '0'
0	00000000	000	0	00000000	NC, Invalid location
1	00000001	000	0	00000001	NC, Insert
2	00000010	001	1	00000001	-1, Insert
3	00000011	001	1	00000100	+1, Insert
4	00000100	010	2	00000100	NC, Insert
5	00000101	010	2	00000101	NC, Insert
6	00000110	011	3	00000101	-1, Insert
7	00000111	011	3	00001000	+1, Insert
8	00001000	100	4	00001000	NC, Insert
9	00001001	100	4	00001001	NC, Insert
10	00001010	101	5	00001001	-1, Insert
11	00001011	101	5	00001100	+1, Insert
12	00001100	110	6	00001100	NC, Insert
13	00001101	110	6	00001101	NC, Insert
14	00001110	111	7	00001101	-1, Insert
15	00001111	111	7	00010000	+1, Insert
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
127	01111111	111	7	10000000	+1, Insert
128	10000000	000	0	10000000	NC, Insert
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
252	11111100	110	6	11111100	NC, Insert
253	11111101	110	6	11111101	NC, Insert
254	11111110	111	7	11111101	-1, Insert
255	11111111	111	7	11111111	NC, Invalid location

Table - II

Decimal Value	Pixel value before insertion of '1'	Extraction of 5^{th}, 6^{th} & 7^{th} by bit extractor	Decimal value of 5^{th}, 6^{th} & 7^{th} bit	Pixel value after insertion of '1'	Change in Pixel value & comment for insertion of '1'
0	00000000	000	0	00000000	NC, Invalid location
1	00000001	000	0	00000010	+1, Insert
2	00000010	001	1	00000010	NC, Insert
3	00000011	001	1	00000011	NC, Insert
4	00000100	010	2	00000100	-1, Insert
5	00000101	010	2	00000110	+1, Insert
6	00000110	011	3	00000110	NC, Insert
7	00000111	011	3	00000111	NC, Insert
8	00001000	100	4	00000111	-1, Insert
9	00001001	100	4	00001010	+1, Insert
10	00001010	101	5	00001010	NC, Insert
11	00001011	101	5	00001011	NC, Insert
12	00001100	110	6	00001011	-1, Insert
13	00001101	110	6	00001110	+1, Insert
14	00001110	111	7	00001110	NC, Insert
15	00001111	111	7	00001111	NC, Insert
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
127	01111111	111	7	01111111	NC, Insert
128	10000000	000	0	01111111	-1, Insert
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
252	11111100	110	6	11111011	-1, Insert
253	11111101	110	6	11111110	+1, Insert
254	11111110	111	7	11111110	NC, Insert
255	11111111	111	7	11111111	NC, Invalid location

6. Results and Conclusion

The following results are obtained from Table I and Table II.

i) The message bit will be inserted at the pseudo random location at first chance

= 508/512*100= 99.21%

ii) Chance when message is inserted, no change in pixel value is required

= 254/508*100= 50%

Our method has several advantages over previous methods like LSB method and GLM method. Firstly, if intruder changes the LSB of all locations then there will be no change on hidden message in our case. Secondly, if LSB of some locations changes due to noise imperfections then hidden message will not be distorted in our method. So, we can say that our method is better than the previous methods like LSB method and GLM method.

7. References

[1] Gutub, M. Faltani, "A Novel Arabic Text Steganography Method Using Letter Points and Extension", WASET International Conference on Computer Information and System Science and Engineering (ICCISSE), Vienna, Austria, May 25-27, 2007.

[2] RJ Anderson, "Stretching the Limit of Steganography", In Information Hiding, Springer Lecture Notes in Computer Science. Vol. 1174, pp 39-48, 1996.

[3] E Franz, A Jerichow, S Moller, A Pfitznaun, I Stierand, "Computer Based Steganography", Information Hiding, Springer Lecture Notes in Computer Science, Vol. 1174, pp 7-21, 1996.

[4] Neil F Johnson, Sushil Jajodia, "Exploring Steganography : Seeing the Unseen", IEEE Computer, pp 26-34, Feb 1998.

[5] GJ Simmons, "The Prisoners Problem and the Subliminal Chownell", Proceedings of crypto'83, Plenum Press, pp 51-67, 1983.

[6] Vidyasagar M. Potdar, Elizabeth Chang, "Gray Level Modification Steganography for Secret Communication", 2nd IEEE International Conference on Industrial Informatics INDIN 2004 June 24th, 26th June, Berlin, Germany, Submitted Tuesday, May 25, 2004.

[7] RJ Anderson, FAP Petitcolas, "On the Limits of Steganography", IEE Journal on selected Areas in Communications, Vol. 16 No 4, pp 474-481, May 1998.

[8] RG Van Schyndel, AZ Tirkel, CF Osborne, "A Digital Watermark", IEEE International Conference on Image Processing, Vol 2, pp 86-90, Nov 1994.

[9] Yeuan-Kuen Lee, Ling-Hwei Chen, "A Secure Robust Image Steganography Model", 10th National Conference on Information Security, Hualien, Taiwan, pp 275-284, May 2000.

[10] Parmender, Sudhir Batra and H R Sharma, "Evaluating the Performance of Message Hidden in First and Second Bit Plane", WSEAS Transaction on Information Science and Technology, Vol. 2, No. 89, pp 1220-1222, Aug 2005.