

Improving the Performance of Least Significant Bit Substitution Steganography against Rs Steganalysis by Minimizing Detection Probability

Vijay Kumar Sharma, Vishal Shrivastava

Arya college of Engineering and Information Technology, Jaipur, Rajasthan ,India

ABSTRACT

Steganography is a branch of information hiding. It embeds the secret message in the cover media (e.g. image, audio, video, etc.) to hide the existence of the message. In recent years, many successful steganography methods have been proposed. With the extensive application of steganography, it is challenged by steganalysis. The most notable steganalysis algorithm is the RS attack which detects the stego-message by the statistic analysis of pixel values. To ensure the security against the RS analysis[1], a new steganographic algorithm is presented based on genetic algorithm in this paper. After embedding the secret message in LSB (least significant bit) of the cover image, the pixel values of the stego image are modified by the genetic algorithm[2] to keep their statistic characters. Thus, the existence of the secret message is hard to be detected by the RS analysis. Meanwhile, better visual quality can be achieved by the proposed algorithm. The experimental results demonstrate the proposed algorithm's effectiveness in resistance to steganalysis with better visual quality.

Keywords: *LSB steganography, PSNR, Zigzag Scan, Flipping functions, RS Steganalysis.*

1. INTRODUCTION

In LSB steganography, the least significant bits of the cover media's digital data are used to conceal the message. The simplest of the LSB steganography techniques is *LSB replacement*. LSB replacement steganography flips the last bit of each of the data values to reflect the message that needs to be hidden. Consider an 8-bit grayscale bitmap image where each pixel is stored as a byte representing a grayscale value. Suppose the first eight pixels of the original image have the following grayscale values:

```
11010010
01001010
10010111
10001100
00010101
01010111
00100110
01000011
```

To hide the letter C whose binary value is 10000011, we would replace the LSBs of these pixels to have the following new grayscale values:

```
11010011
01001010
10010110
10001100
00010100
01010110
00100111
01000011
```

Note that, on average, only half the LSBs need to change. The difference between the cover (i.e. original) image and the stego image will be hardly noticeable to the human eye.

Figure 1.1(a),(b) that show a cover image and a stego image (with the message "hello world" embedded); there is no visible difference between the two images.



Fig: 1.1 (a) Cover Image



Fig: 1.2 (b) Stego Image

LSB steganography, as described above, replaces the LSBs of data values to match bits of the message. It can equally alter the data value by a small amount, ensuring the a legal range of data values is preserved. The difference being that the choice of whether to add or subtract one from the cover image pixel is random. This will have the same effect as LSB replacement in terms of not being able to perceive the existence of the hidden message. This steganographic technique is called *LSB matching*. Both LSB replacement and LSB matching leave the LSB unchanged if the message bit matches the LSB. When the message bit does not match the LSB, LSB replacement replaces the LSB with the message bit; LSB matching randomly increments or decrements the data value by one. LSB matching is also known as ± 1 embedding.

In the case of still grayscale images of type bitmap, every pixel is represented using 8 bits, with 11111111 (=255) representing white and 00000000 (=0) representing black. Thus, there are 256 different grayscale shades between black and white which are used in grayscale bitmap images. In LSB steganography, the LSB's of the cover image is to be changed. As the message bit to be substituted in the LSB position of the cover image is either 0 or 1, one can state without any loss of generality that the LSB's of about 50 percent pixel changes.

There are three possibilities [3]:

1. Intensity value of any pixel remains unchanged.
2. Even value can change to next higher odd value

Odd Value change to previous lower even value

2. RS steganalysis

Steganalysis refers to the analysis of any image with an objective of trying to find out whether the image is a stego image or not. RS steganalysis is based on Statistical techniques that starts with the sample counts. It can be proved mathematically that utilizing the spatial correlations in the stego-image, one should be able to build much more reliable and accurate detection.

RS Analysis partitions an image into groups consisting of 'n' adjacent pixels, and computes the "smoothness" of each group using a discrimination function f . The discrimination function for pixel values x_1, x_2, \dots, x_n of a group G is defined as follows.

$$f(G) = f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n |x_{i+1} - x_i| \quad (2.1)$$

where x is the pixel value and n is the number of pixels.

In RS steganalysis, 3 kinds of block flipping are defined. They are positive flipping F_1 , negative flipping F_{-1} and 0 flipping F_0 . F_1 is the transformation relationship between $2i$ and $2i+1$, (i.e. $0-1, 2-3, \dots, 254-255$), which is same as LSB. F_{-1} is the transformation relationship between $2i-1$ and $2i$, i.e. $-1-0, 1-2, \dots, 255-256$.

The relationship between the two flipping is written as:

$$F_{-1} = F_1(x + 1) - 1 \quad (2.2)$$

Similarly, define F_0 as the identity permutation

$$F_0(x) = x \quad (2.3)$$

F_0, F_1 and F_{-1} are called flipping functions. The flipped group is resulted from applying flipping functions on pixels of image block. It is denoted as:

$$F(G) = (F_{M(1)}(x_1), F_{M(2)}(x_2), \dots, F_{M(n)}(x_n)) \quad (2.4)$$

$M = M(1), M(2), \dots, M(n)$ is called a flipping mask, where $M(i)$ are 1, 0, or -1 . G is regular if $f(F(G)) > f(G)$, G is singular if $f(F(G)) < f(G)$.

The RS analysis includes following steps. Firstly, the image is divided into non-overlapping blocks and each one is re-arranged into a vector $G = (x_1, x_2, \dots, x_n)$ in the Zigzag scan order. The correlations of pixels can be determined by discrimination function.

The value of f represents the spatial correlation between the adjacent pixels. A small f means the strong correlation.

After all the $f(G)$ are obtained, apply non-negative flipping (i.e. $M(1), M(2), \dots, M(n) = 0$ or 1) and non-positive flipping (i.e. $M(1), M(2), \dots, M(n) = 0$ or -1) on each block.

Then use Eq.2.1 to calculate $f(F(G))$ in each block. The relative number of regular blocks after positive flipping is denoted as R_m , and that of singular blocks is denoted as S_m .

In the same way, R_{-m} and S_{-m} are defined as the relative number of regular and singular blocks after the negative flipping.

It is pointed out by Fridrich [3] that in nature images, the numbers of aforementioned blocks hold the following relationships:

$$R_m \approx R_{-m}, S_m \approx S_{-m} \text{ and } R_m > S_m, R_{-m} > S_{-m} \quad (2.5)$$

The LSB randomization (which is what happens with LSB replacement) reduces the difference between R_M and S_M to close to zero, and increases the difference between R_{-m} and S_{-m} .

The difference between R_m and R_{-m} increases with the length of the embedded message. The same trend exists in the difference between S_m and S_{-m} . Consider the grayscale image block as shown in fig 2.1 below:

B=	107	109	107	105	104	102	102	104
	107	106	105	104	105	103	105	102
	107	105	107	105	102	103	104	103
	107	107	105	106	104	103	103	104
	107	109	107	104	104	102	103	102
	104	107	106	103	103	104	102	100
	110	109	109	105	105	105	105	102

Fig: 2.1 A 8 X 8 grayscale image Block

A Zigzag scan of this image block can be done as shown in fig 2.2[2]:

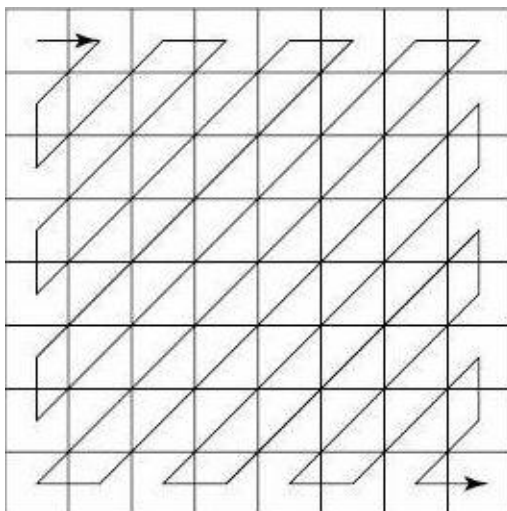


Fig:2.2 Zigzag scan of a 8 x 8 grayscale image block

For the above Block B Discrimination Function $f=99$.

The RS analysis performs random flippings several times on the 8X8 image blocks and calculates R_+ , R_- , S_+ and S_- blocks. The presence of the stego image can be justified by the condition that the number of R-blocks increases significantly.

The primary motivation of the current work is to reduce the number of R- blocks.

It is pointed out by Shen Wang and others that the type- regular and singular of a block can be changed by changing the pixel values in such a way that the distortion in the image is minimal.

As the higher order bits significantly change the intensity value of the pixel, the aim is to find out a set of pixel values by changing the second LSB's so that the resultant block will be singular.

After finding out the label for a particular block of the stego image, as done in RS analysis, we proceed to operate all those blocks which are of type R-.

For this purpose we employ the Genetic Approach: Algorithm:

1. Initialization:

Starting from the first pixel, select two pixels adjacent to each other in a row as shown in the figure. These are known as the initial chromosomes in the genetic terminology:

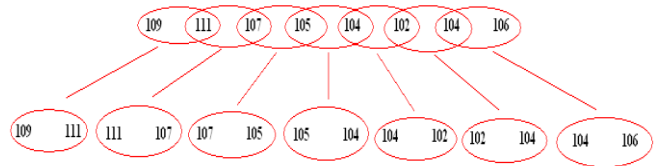


Fig: 2.3 First Generation Chromosomes

2. Reproduction and Mutation:

Flip the second lowest bits in the chromosomes randomly to generate the four possible chromosomes of the next generation as shown:

First Pixel Value	Second Pixel Value	Second LSB Operation	Next Generation Chromosomes
109	111	No Flip	C1 109 111
		Flip	C2 107 111
		No Flip	C3 109 113
		Flip	C4 107 113

3. Selection:

Select the best possible chromosome which maximizes the value of the flipping function $F = \alpha \cdot \text{Prob}[f-(C_i) < f(C_i)] + \beta \cdot \text{PSNR}$ (2.6) Where α and β are positive constants. Also the Probability $\text{Prob}[f-(C_i) < f(C_i)]$ must be greater than some threshold.

4. Calculate the label of the adjusted image block using genetic algorithm. The block is successfully adjusted.

5. Crossover. Shift the chromosomes one pixel, goto step 2. If crossover has been applied two times, stop the cycle.
6. After a block is adjusted, calculate R , R^- , S and S^- of the image. If the difference between R and R^- is more than 5%, or the the difference between S and S^- is more than 5%, adjust the next block.

In this algorithm, the blocks are labeled before the adjustment. Thus, the computational complexity is reduced. The usage of the genetic method avoids the exhausting searching and the algorithm is easy to be implemented.

3. EXPERIMENTAL RESULTS OF THE ALGORITHM

3.1 Description of Simulator

The Steganographic algorithm presented in present work is implemented using PHP and MySQL database. PHP is a server side scripting language having a rich set of tools to manipulate images of all types, e.g. jpeg, gif, png etc.

The code of the simulator along with the comments is enclosed in the appendix for ready reference. Following scripts are used for the implementation of the algorithm.

Database: **Steganography**

The database consists of two tables. One table consists of the decimal values of pixels of original image while the other table stores the modified pixel values of the stego image.

Package: **StegoImage**

1. **color_to_greyscale.php**: To convert the jpeg color image to greyscale. It takes the sum of three RGB values, calculate the average and sets the resultant 8 bits which then displays the 8 bit greyscale shade from the greyscale color palette. It also saves the data into pixel_values table.
2. **image_matrix.php**: This script fetches the data from pixel_values table. It displays the matrix of the dimensions equal to the height and width of the image, in which each cell consist of the value (can be set to binary, octal, decimal, hex etc.) which corresponds to the intensity of the pixel of those coordinates.
3. **secret_message.php**: It takes a character string (secret message) and converts the same into binary string. It then displays the matrix of the dimensions equal to the height and width of the image, in which each cell consist of the value (can be set to binary, octal, decimal, hex etc.)

which corresponds to the modified value of the intensity of the pixels due to LSB substitution. The number of pixels modified equals the length of the secret message. The LSB of the pixels of the original image are then replaced by the bits of the secret message. It also saves the data into modified_pixel_values table.

4. **stego_image.php**: This script fetches the data from modified_pixel_values table. It displays the Stego Image.
5. **image_extract.php**: this script retrieves the message from the stego image by retrieving the LSB's of all the pixels in the stego image, reading the image matrix in row major order.

3.2 Experimental Results of the Simulator

The original image and its 8 times magnified view with pixel grid is shown in Figures 3.1 and 3.2 respectively. Followed by the experimental results of the simulator.



Fig. 3.1 Original Image



Fig 3.2 Original Image, Magnified 8 times in pixel grid view

3.2.1 Conversion of Color image into Greyscale Image

Conversion of a color image to grayscale can be done using several approaches. Different weighting of the

primary colors effectively represents the effect of obtaining black-and-white image with color images. A common strategy is to match the luminance of the grayscale image to the luminance of the color image.

To convert any color to a grayscale representation[4][5] of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.), the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation.

To convert a gray intensity value to RGB, simply set all the three primary color components red, green and blue to the gray value, correcting to a different gamma if necessary.

The method adopted in current work for experimental evaluation is to obtain the RGB values of individual pixels and to take the average to be normalized to fit in the scale 0 to 255.

The 8 bit grayscale color palette is as shown in the figure below.

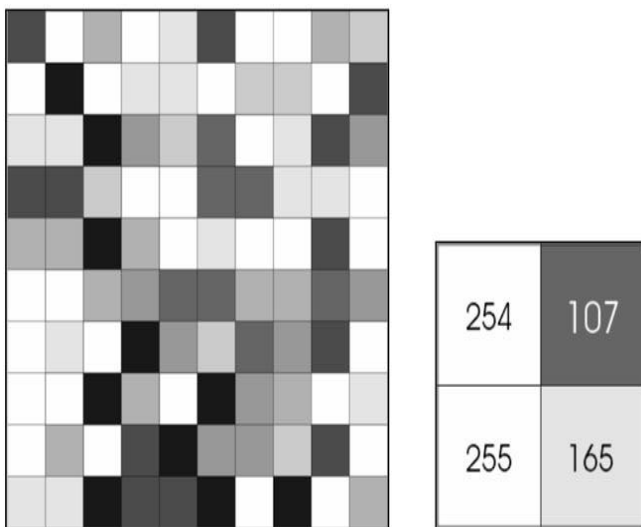


Fig. 3.3 8 Bit Grayscale Color Palette

The grayscale image obtained from the color image and its magnified view with pixel grid are shown in Figure 3.4 and 3.5 respectively.



Fig. 3.4 GreyScale Image obtained by normalizing RGB values

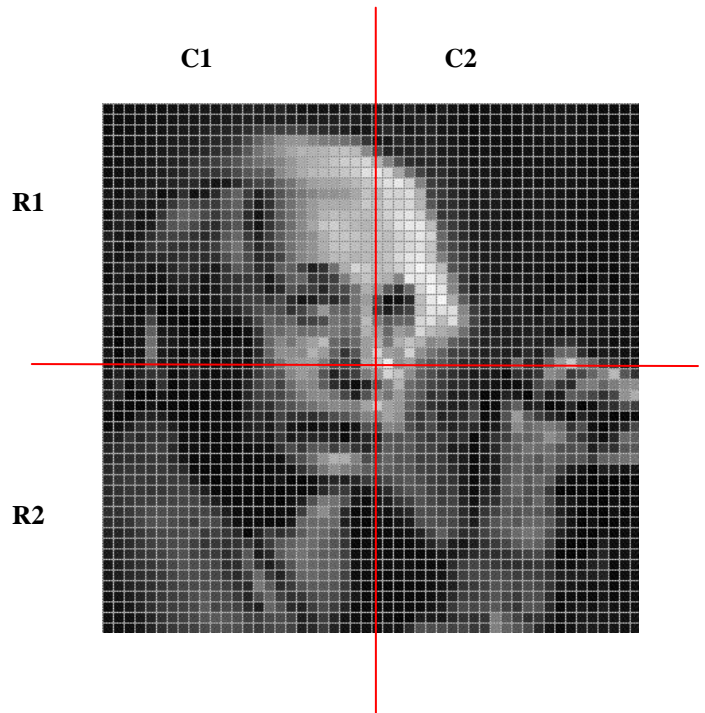


Fig. 3.5 Original Image, Magnified 8 times in pixel grid view

Hall Figure is divided in to four parts, these are as follow:

1. $C1 * R1$
2. $C2 * R1$
3. $C1 * R2$
4. $C2 * R2$
- 5.

3.3 Pixel Matrix

The pixel matrix, in decimal representation, obtained as output from the simulator can be represented as shown in figure 3.6,3.7,3.8 and 3.9

The Image Matrix for the greyscale image is as shown below:

```

12 12 12 10 10 12 15 16 14 20 22 19 27 25 29 22 16 23 29 26 18 18 21 21 24
16 16 16 16 16 17 18 19 24 25 26 35 27 28 22 12 23 16 21 32 28 15 6 8 7
16 16 17 19 18 16 18 21 22 29 31 33 26 13 18 45 30 14 9 15 12 6 10 21 23
16 15 17 20 20 18 21 26 31 31 36 8 14 38 49 21 21 15 13 14 18 22 23 20 17
19 18 20 23 24 24 28 33 35 21 16 31 60 59 30 23 20 17 18 31 62 98 11210892
19 19 21 23 25 29 33 36 20 22 48 80 63 61 34 34 18 17 16 14 15 25 34 35 27
18 21 24 24 29 35 37 35 27 61 97 74 67 58 43 37 36 20 9 9 18 26 23 14 11
22 27 30 31 35 41 42 37 47 59 12067 79 41 60 52 42 35 23 9 7 15 22 20 22
25 22 33 29 42 41 61 45 16 74 11253 60 48 57 43 35 20 22 15 9 7 16 27 33
20 30 24 40 42 51 68 20 30 51 80 51 47 39 39 31 23 24 22 11 16 16 13 15 15
22 27 30 43 43 73 53 17 53 52 52 42 51 66 48 26 21 28 28 17 16 17 13 16 17
27 28 36 42 54 79 45 38 76 86 75 76 94 91 77 61 49 39 40 35 13 13 15 20 17
28 38 35 53 74 78 69 50 94 11692 77 10110211788 56 42 40 41 16 16 15 12 17
31 32 48 74 88 99 89 55 58 49 30 41 63 55 83 60 46 61 67 62 28 23 14 22 20
32 29 58 10310412091 64 30 30 23 28 29 39 57 49 49 51 51 68 57 60 33 28 36
26 44 47 12812311593 73 47 54 49 56 49 61 39 43 30 26 48 97 90 10710110684
31 40 56 14314613913297 85 88 92 85 91 83 79 79 58 40 59 86 79 119135138128
27 37 60 15316315915311811010611310511912010610095 83 60 43 94 107116146144
21 28 57 15117217417414211912914113713313012270 91 55 20 80 67 107137141151
21 24 49 13817318118516013215616216914814715640 61 65 22 98 62 113173171169
    
```



25 26 47 12917918919016614616917919518017516936 49 45 49 14584 123174164135
 22 26 46 12118819919016515217119319920019918512669 71 59 10310813710410369
 20 27 37 10519020719517315116919719319719218118012753 63 85 10210190 52 56
 23 28 30 91 18721220418816818019018619519219820117613210513211610810083 56
 27 28 26 49 146215216201191192191178194192187201194172158162205170158154169

Fig 3.6: Matrix representing the GreyScale Image Pixel Values of C1* R1 in Decimal (Dimension 25X25)

21 31 36 22 9 15 25 38 39 42 44 45 47 46 42 32 28 29 28 22 18 17 14 14 14
 20 35 29 11 7 22 33 38 41 52 53 44 48 52 43 37 32 33 35 31 28 25 21 16 16
 25 25 13 2 14 33 42 46 45 58 63 52 54 62 54 50 45 43 43 40 36 30 23 20 18
 27 26 10 6 27 47 48 58 46 51 61 58 60 68 70 58 54 51 48 45 41 37 33 27 23
 48 9 5 21 35 44 51 60 56 52 52 56 62 70 78 60 60 57 53 50 49 49 50 38 33
 54 50 14 7 37 55 45 52 64 61 51 55 66 74 81 68 72 70 63 58 55 55 58 51 45
 11 13 14 11 11 28 51 36 47 56 58 61 68 75 78 74 80 80 74 69 63 58 60 61 54
 21 15 8 9 15 18 19 19 17 36 60 66 65 69 72 67 76 80 79 78 73 67 66 65 58
 41 26 10 12 12 10 14 13 12 10 26 58 68 65 70 76 60 80 78 69 81 80 80 74 63
 14 14 12 9 11 11 7 10 8 8 12 28 53 70 72 69 52 64 72 78 86 80 85 84 68
 14 14 15 10 9 13 15 9 9 8 7 11 33 58 68 61 65 54 65 68 64 81 74 88 74
 20 17 13 16 10 6 8 9 9 12 9 9 14 16 27 48 56 50 49 51 67 69 63 86 88 81
 22 19 13 14 16 13 13 8 9 7 9 11 7 9 21 34 70 54 39 51 48 49 64 78 78
 21 26 27 25 32 32 15 14 7 8 8 4 9 15 8 15 33 92 84 41 45 45 36 66 73
 56 83 81 52 37 39 37 20 11 11 10 5 11 17 9 39 31 16 66 125 113 72 40 56 57
 8173 63 51 36 23 20 19 12 8 11 14 6 5 9 59 76 41 25 53 90 110 94 76 64
 89 68 64 72 74 51 55 48 24 12 14 4 13 6 11 39 105 80 49 45 37 62 85 83 72
 128 116 101 73 38 28 28 70 51 15 6 11 8 10 30 59 105 109 97 75 51 47 30 66 68
 157 132 125 103 51 10 37 57 89 42 8 14 17 86 83 99 105 108 109 99 88 71 35 28 36
 135 137 132 94 64 16 32 86 105 78 19 6 83 117 107 102 102 105 101 96 88 71 62 48 47
 76 100 145 124 63 15 42 85 153 109 40 18 147 129 106 103 103 107 103 100 91 73 73 62 54
 36 28 133 131 76 13 29 70 186 131 54 56 102 106 114 91 82 89 95 94 90 75 59 41 35
 85 19 64 128 104 16 20 73 158 131 34 70 73 39 24 51 54 68 76 65 58 53 37 21 21
 94 142 138 75 7 29 107 86 117 20 3 9 10 9 12 13 14 14 14 15 13 12 16 21 21
 38 145 184 62 11 73 90 88 120 23 5 11 12 12 14 14 16 17 17 17 14 14 17 19 20

Fig 3.7: Matrix representing the GreyScale Image Pixel Values of C1* R1 in Decimal (Dimension 25X25)

20 29 28 38 59 192 228 216 201 195 200 180 189 198 190 186 191 146 78 59 131 145 188 217 207
 24 28 25 33 35 122 227 216 204 214 207 200 195 184 203 193 169 96 28 30 85 112 96 162 230
 22 29 34 26 47 62 192 217 207 212 225 218 212 199 202 213 148 29 22 44 100 174 172 146 156
 28 21 33 34 42 38 112 217 220 224 222 238 236 216 236 224 222 107 38 83 83 148 196 208 165
 20 32 30 34 35 44 66 128 177 178 214 223 221 228 240 233 225 223 192 189 217 208 176 156 108
 22 33 21 36 45 36 45 52 74 77 130 201 220 184 199 238 247 240 231 230 216 199 134 113 83
 20 27 27 29 26 40 38 51 52 54 62 66 103 106 120 182 208 230 241 224 197 173 145 101 87
 20 22 23 23 26 31 36 37 42 45 52 61 66 64 71 84 123 139 182 224 215 154 97 73 63
 14 15 15 15 16 20 24 25 30 32 36 42 46 45 49 57 56 57 85 134 155 126 78 48 53
 18 18 17 17 19 22 25 25 25 26 26 28 32 32 32 34 39 36 42 55 57 49 43 44 25
 17 16 16 16 19 24 26 26 27 29 30 28 27 28 31 36 38 35 33 35 37 33
 14 14 14 14 18 22 24 23 24 27 26 24 25 26 25 25 28 29 29 28 32 35 33 31 41
 17 17 17 17 19 22 23 23 21 24 23 22 23 23 22 24 18 24 29 28 27 29 34 37 32
 16 17 17 16 16 18 20 20 22 23 22 22 23 22 21 23 25 25 27 28 30 32 34 34 91
 17 19 20 19 18 20 22 23 24 22 20 21 24 22 20 22 20 20 24 30 33 35 37 39 46
 20 20 20 19 17 18 20 21 24 24 24 25 24 23 22 21 30 19 28 35 29 39 37 43
 22 22 22 21 20 20 21 23 23 24 24 25 26 25 23 21 11 13 25 27 24 29 38 44 93
 20 21 22 21 20 19 20 21 19 20 21 23 23 21 19 17 21 16 28 28 37 30 32 100 166
 19 21 22 21 20 19 19 20 18 18 19 20 20 19 17 14 17 14 22 19 28 20 35 125 199
 19 20 22 22 20 19 19 20 20 19 18 18 19 18 17 15 16 9 19 21 25 33 30 34 118
 14 15 17 17 16 16 16 17 17 16 15 15 15 15 14 13 12 15 19 21 22 26 28 21 108
 11 12 14 14 14 13 14 16 15 15 14 14 14 13 11 13 18 16 20 27 17 15 28 56
 14 16 17 17 17 19 20 19 19 19 19 18 16 14 11 13 24 21 18 24 24 21 37
 14 15 16 17 17 17 18 19 19 18 17 17 18 17 16 15 15 17 19 19 18 19 18 17
 13 14 15 16 16 16 16 17 16 17 16 15 15 16 15 13 14 14 17 18 17 17 18 17 19

Fig 3.8: Matrix representing the GreyScale Image Pixel Values of C1* R2 in Decimal (Dimension 25X25)

63 179 188 132 89 25 32 99 96 47 39 32 13 16 9 11 12 11 12 11 11 14 13 12
 138 100 173 162 120 80 103 96 94 76 40 23 20 11 12 13 13 11 9 7 7 9 15 10
 190 159 136 152 119 95 92 84 88 89 68 46 35 25 13 13 13 13 10 8 8 9 11 18
 120 98 99 71 86 79 80 82 75 77 85 67 36 22 17 10 10 14 13 13 15 16 11 46
 92 72 71 61 67 71 69 74 81 74 68 84 92 71 38 18 9 12 13 11 12 12 27 85
 73 66 59 51 57 59 63 59 57 65 70 70 75 86 69 36 15 15 14 12 14 16 21 79
 58 47 41 42 50 57 56 61 59 56 60 62 67 76 91 48 21 21 22 23 29 35 30 75
 49 30 41 31 39 35 31 38 49 57 60 64 68 70 71 26 30 38 46 32 45 65 44 72
 42 33 25 40 55 48 26 19 16 19 27 37 48 53 47 19 37 63 70 50 58 65 54 73
 20 34 59 75 33 17 17 12 10 15 20 24 26 27 43 37 45 55 69 68 71 64 47 55
 19 12 5 17 10 18 24 23 22 24 30 38 52 67 65 51 54 67 77 59 56 84 60 61
 16 9 12 24 27 49 38 36 31 31 36 45 64 86 84 48 44 60 81 83 73 96 125 127
 22 31 70 85 60 66 72 74 77 84 89 86 86 96 96 68 54 53 55 79 71 63 92 102
 15 31 76 154 137 104 101 98 93 94 95 85 79 86 107 83 46 58 65 70 61 48 66 91
 16 12 21 109 99 96 108 101 88 85 86 83 88 105 109 120 53 52 65 71 79 79 43 79
 82 62 27 65 118 105 99 93 84 90 100 95 92 100 103 116 71 50 58 70 76 74 62 37
 78 35 15 19 125 146 120 100 100 83 56 66 74 85 78 51 28 58 54 57 41 45 29 15
 60 21 11 22 125 104 77 90 64 32 13 25 37 35 46 4 10 16 34 18 16 9 17 15
 52 91 27 16 43 84 83 65 39 13 5 25 13 23 9 5 7 12 32 25 16 13 15 19
 111 90 19 25 65 66 83 98 17 4 3 9 5 6 3 9 5 21 39 13 12 23 26 31
 139 49 23 25 26 18 37 46 6 8 2 9 3 5 10 12 13 21 41 12 17 17 26 29
 89 57 36 9 6 15 48 107 18 6 2 0 6 2 5 39 44 51 41 18 17 23 25 25
 120 86 31 7 11 44 48 117 20 2 3 8 8 8 12 13 9 9 9 18 11 20 24 20
 142 138 75 7 29 107 86 117 20 3 9 10 9 12 13 14 14 14 15 13 12 16 21 21
 145 184 62 11 73 90 88 120 23 5 11 12 12 14 14 16 17 17 17 14 14 17 19 20

Fig 3.9: Matrix representing the GreyScale Image Pixel Values of C2* R2 in Decimal (Dimension 25X25)

Consider the Hidden Message:

“The true logic of this world lies in the calculus of Probabilities: James Clark Maxwell”

The conversion of the above hidden message in binary string can be done as per the following scheme:

TABLE 3.1 CHARACTERS WITH ASCII VALUES, FOLLOWED BY CONVERSION INTO 8 BIT BINARY NUMBERS

S. No	Character	ASCII value	Binary Representation
1	T	84	01010100
2	h	104	01101000
3	e	101	01100101
4	[space]	32	00100000
5	t	116	01110100
6	r	114	01110010
7	u	117	01110101
8	e	101	01100101
9	[space]	32	00100000
10	l	108	01101100
11	o	111	01101111
12	g	103	01100111
13	i	105	01101001
14	c	99	01100011
15	[space]	32	00100000
16	o	111	01101111
17	f	102	01100110
18	[space]	32	00100000
19	t	116	01110100



20	h	104	01101000
21	i	105	01101001
22	s	115	01110011
23	[space]	32	00100000
24	w	119	01110111
25	o	111	01101111
26	r	114	01110010
27	l	108	01101100
28	d	100	01100100
29	[space]	32	00100000
30	l	108	01101100
31	i	105	01101001
32	e	101	01100101
33	s	115	01110011
34	[space]	32	00100000
35	i	105	01101001
36	n	110	01101110
37	[space]	32	00100000
38	t	116	01110100
39	h	104	01101000
40	e	101	01100101
41	[space]	32	00100000
42	c	99	01100011
42	a	97	01100001
44	l	108	01101100
45	c	99	01100011
46	u	117	01110101
47	l	108	01101100
48	u	117	01110101
48	s	115	01110011
50	[space]	32	00100000
51	o	111	01101111
52	f	102	01100110
53	[space]	32	00100000
54	P	80	01010000
55	r	114	01110010
56	o	111	01101111
57	b	98	01100010
58	a	97	01100001
58	b	98	01100010
60	i	105	01101001
61	l	108	01101100
62	i	105	01101001
63	t	116	01110100
64	i	105	01101001
65	e	101	01100101
66	s	115	01110011
67	:	58	00111010
68	[space]	32	00100000
69	J	74	01001010
70	a	97	01100001
71	m	109	01101101
72	e	101	01100101
73	s	115	01110011
74	[space]	32	00100000
75	C	67	01000011

76	l	108	01101100
77	a	97	01100001
78	r	114	01110010
79	k	107	01101011
80	[space]	32	00100000
81	M	77	01001101
82	a	97	01100001
83	x	120	01111000
84	w	119	01110111
85	e	101	01100101
86	l	108	01101100
87	l	108	01101100

Total characters: 87
Total Bits: 696 (=87X8)

The complete binary string for the secret message is:

```
01010100011010000110010100100000011101000111001
00111010101100101001000000110110001101111011001
11011010010110001100100000011011110110011000100
00001110100011010000110100101110011001000000111
01110110111101110010011011000110010000100000011
01100011010010110010101110011001000000110100101
10111000100000011101000110100001100101001000000
11000110110000101101100011000110111010101101100
01110101011100110010000001101111011001100010000
00101000001110010011011110110001001100001011000
10011010010110110001101001011101000110100101100
10101110011001110100010000001001010011000010110
11010110010101110011001000000100001101101100011
00001011100100110101100100000010011010110000101
11100001110111011001010110110001101100
```

3.4 Simulator modeling LSB and RS Steganalysis

The Modified Image Matrix is as shown in figure 3.10,3.11,3.12 and 3.13. The underlined pixel values are those that are changed. The change occurs in total 696 pixels.

12	<u>13</u>	12	11	10	<u>13</u>	<u>14</u>	16	14	<u>21</u>	<u>23</u>	<u>18</u>	27	<u>24</u>	<u>28</u>	22	16	23	29	26	18	<u>19</u>	<u>20</u>	21	24
<u>17</u>	<u>16</u>	<u>17</u>	16	17	18	19	<u>25</u>	<u>24</u>	26	35	<u>26</u>	<u>29</u>	22	12	23	16	<u>20</u>	32	28	<u>14</u>	6	9	7	
<u>17</u>	<u>16</u>	19	18	<u>17</u>	<u>19</u>	<u>20</u>	22	<u>28</u>	31	33	26	<u>12</u>	<u>19</u>	<u>44</u>	30	14	8	<u>14</u>	12	7	<u>11</u>	<u>20</u>	23	
<u>16</u>	<u>14</u>	<u>16</u>	<u>21</u>	<u>18</u>	21	26	<u>30</u>	<u>30</u>	36	9	15	38	49	<u>20</u>	<u>15</u>	<u>12</u>	<u>15</u>	<u>19</u>	<u>23</u>	<u>22</u>	20	17		
<u>18</u>	<u>19</u>	<u>21</u>	<u>23</u>	24	<u>29</u>	<u>32</u>	<u>34</u>	21	<u>17</u>	<u>30</u>	<u>61</u>	59	30	22	20	17	<u>19</u>	<u>30</u>	<u>62</u>	<u>99</u>	112	108	92	
<u>19</u>	<u>18</u>	<u>20</u>	<u>23</u>	<u>24</u>	<u>29</u>	<u>32</u>	<u>37</u>	<u>21</u>	<u>23</u>	48	80	63	61	34	34	<u>19</u>	<u>16</u>	16	14	<u>14</u>	<u>24</u>	<u>34</u>	35	27
18	21	24	24	<u>28</u>	35	37	<u>34</u>	27	<u>60</u>	<u>96</u>	74	<u>66</u>	<u>59</u>	43	<u>36</u>	<u>36</u>	<u>21</u>	8	9	18	26	23	14	<u>10</u>
<u>22</u>	<u>26</u>	30	31	35	<u>40</u>	<u>42</u>	<u>36</u>	47	59	120	67	79	41	60	<u>53</u>	<u>42</u>	35	<u>22</u>	9	7	<u>14</u>	<u>23</u>	<u>21</u>	22
<u>24</u>	<u>23</u>	<u>33</u>	<u>28</u>	<u>43</u>	41	61	45	16	<u>75</u>	<u>113</u>	<u>52</u>	60	<u>49</u>	57	<u>42</u>	<u>34</u>	<u>20</u>	<u>23</u>	<u>14</u>	8	6	16	<u>26</u>	<u>32</u>
21	30	24	40	<u>43</u>	<u>50</u>	68	21	31	<u>50</u>	<u>80</u>	<u>50</u>	<u>46</u>	39	<u>38</u>	31	23	24	22	<u>10</u>	<u>17</u>	<u>16</u>	<u>12</u>	15	15
22	27	30	<u>42</u>	<u>42</u>	73	53	<u>16</u>	53	52	<u>52</u>	<u>43</u>	<u>50</u>	<u>67</u>	<u>49</u>	<u>26</u>	<u>20</u>	<u>29</u>	28	17	16	17	13	<u>17</u>	<u>16</u>
27	28	36	<u>43</u>	<u>55</u>	<u>78</u>	<u>44</u>	38	76	<u>87</u>	<u>74</u>	<u>77</u>	<u>95</u>	<u>90</u>	77	61	<u>48</u>	39	40	35	13	<u>12</u>	<u>14</u>	<u>21</u>	<u>16</u>
28	<u>39</u>	35	<u>52</u>	<u>75</u>	<u>79</u>	<u>68</u>	50	94	<u>117</u>	<u>93</u>	<u>76</u>	<u>100</u>	<u>102</u>	<u>116</u>	<u>89</u>	56	<u>43</u>	<u>41</u>	41	16	16	15	<u>16</u>	<u>16</u>
31	32	48	74	88	99	<u>88</u>	55	<u>59</u>	49	<u>31</u>	<u>40</u>	<u>62</u>	<u>54</u>	<u>82</u>	<u>61</u>	<u>47</u>	61	<u>66</u>	<u>63</u>	<u>29</u>	<u>23</u>	14	<u>23</u>	<u>21</u>
32	29	58	103	104	120	91	64	30	30	23	28	29	39	57	49	49	51	51	68	57	60	33	28	36
26	44	47	128	123	115	93	73	47	54	49	56	49	61	39	43	30	26	48	97	90	107	101	106	84
31	40	56	143	146	139	132	97	85	88	92	85	91	83	79	79	58	40	59	86	79	119	135	138	128



4. CONCLUSION

The algorithm proposed in the current work describes a method such that the stego image which is obtained thereby, cannot be proved as stego image using the RS steganalysis approach. In the proposed algorithm, the blocks are labeled before the adjustment. Thus, the computational complexity is reduced. The usage of the genetic method avoids the exhausting searching and the algorithm is easy to be implemented.

Moreover, only the selected image blocks are operated, and second most LSB's are changed selectively, one can state qualitatively that the difference between the LSB stego image and the one obtained by block flipping is minimal to be perceived by human eye.

A secure stego algorithm based on the genetic method is proposed in this research. Benefited from the effective optimization, a good balance between the security and the image quality is achieved. Our future work will focus on improving the efficiency of the proposed algorithm. The result of RS steganalysis and PSNR with different genetic parameters are derived and results are tabulated.

REFERENCES

- [1] J. Fridrich, Multimedia Security Technologies for Digital Rights Management. Academic Press, 2006, ch. Steganalysis, pp. 349–381. J. Fridrich, R. Du, and M. Long, “Steganalysis of LSB encoding in color images,” in Proceedings of the IEEE International Conference on Multimedia and Expo. New York, USA: IEEE Computer Society Press, 2000.
- [2] Shen Wang, Bian Yang and Xiamu Niu “A Secure Steganography Method based on Genetic Algorithm” Journal of Information Hiding and Multimedia Signal Processing c 2010 ISSN 2073-4212. Volume 1, Number 1, January 2010
- [3] J. Fridrich, R. Du, and L. Meng, “Steganalysis of LSB Encoding in Color Images,” Proc. IEEE Int'l Conf. Multimedia and Expo, CD-ROM, IEEE Press, Piscataway, N.J., 2000.
- [4] Finlayson, G. D., Qiu, G., Qiu, M., Contrast Maximizing and Brightness Preserving Color to Grayscale Image Conversion, 1999.
- [5] Tarun Kumar, Karun Verma, “A Theory Based on Conversion of RGB image to Gray image”, International Journal of Computer Applications Volume 7– No.2, September 2010.