Reversible Data Hiding Scheme Based on 3-Least Significant Bits and Mix Column Transform

Wafaa Mustafa Abduallah[1], Abdul Monem S. Rahma[2], and Al-Sakib Khan Pathan[1]

[1] Department of Computer Science, International Islamic University Malaysia
Gombak, Kuala Lumpur, 53100, Malaysia
[2] Department of Computer Science, University of Technology, Baghdad, Iraq
{heevy9,monem.rahma}@yahoo.com, sakib@iium.edu.my

Abstract. Steganography is the science of hiding a message signal in a host signal, without any perceptual distortion of the host signal. Using steganography, information can be hidden in the carrier items such as images, videos, sounds files, text files, while performing data transmission. In image steganography field, it is a major concern of the researchers how to improve the capacity of hidden data into host image without causing any statistically significant modification. In this work, we propose a reversible steganography scheme which can hide large amount of information without affecting the imperceptibility aspect of the *stego-image* and at the same time, it increases the security level of the system through using different method for embedding based on distinct type of transform, called Mix Column Transform. Our experimental results prove the ability of our proposed scheme in balancing among the three critical properties: capacity, security, and imperceptibility.

Keywords: Data, Hiding, Mix Column Transform, Polynomial, Steganography.

1 Introduction

Steganography is considered a science or art of secret communication. In the recent years, digital steganography has become a hot research issue due to the wide use of the Internet as a popular communication medium. The goal of digital steganography is to conceal covert message in digital material in an imperceptible manner. Even though digital images, audio files, video data and all types of digital files can be considered as a cover item to conceal secret information, in this paper, we consider only digital images as cover item. After hiding a secret message into the *cover image*, we get an image with secret message; so-called *stego-image*, which is transmitted to a receptor via popular communication channels or put on some Internet website. To design useful steganography algorithm, it is very important that the *stego-image* does not have any visual artifact and it is statistically similar to natural images. If a third party or observer has some suspicion over the *stego-image*, steganography algorithm becomes useless [1]. Three common requirements can be used to rate the performance of steganographic techniques, which are: security, capacity, and imperceptibility [2].

T. Zia et al. (Eds.): SecureComm 2013, LNICST 127, pp. 405–417, 2013.

- *Security:* Many active or passive attacks could be launched against steganography. Hence, if the existence of the secret message can only be estimated with a probability not higher than "*random guessing*" when any steganalytic system is applied, then this steganography may be considered secure under such steganalytic system. Otherwise, we may claim it as insecure.
- *Capacity:* Capacity is a critical aspect of any steganography. The hiding capacity provided by any steganographic scheme should be as high as possible, which may be given with absolute measurement (e.g., the size of secret message), or with relative value (e.g., data embedding rate, such as bits per pixel, bits per non-zero discrete cosine transform coefficient, or the ratio of the secret message to the cover medium, etc.).
- *Imperceptibility*: *Stego images* should not have severe visual artifacts. Under the same level of security and capacity, the higher the fidelity of the *stego image* is, the better it is. If the resultant *stego image* appears innocuous enough, one can believe this requirement to be satisfied well for the possessor not having the original *cover image* to compare.

Steganography can be mainly classified into four categories: (1) Steganography in image, (2) Steganography in audio, (3) Steganography in video, and (4) Steganography in text. The image steganography algorithms can be divided into two categories, namely, spatial domain and frequency domain [3]. In this work, a distinct type of transform will be applied on the color image called "Mix Column Transform" (MCT) based on some different type of mathematics called irreducible polynomial mathematics, which can meet the requirements of good steganographic system (high capacity, good visual imperceptibility, and reasonable level of security).

After Section 1, in Section 2, we discuss the related works and our motivation for this work. The mathematical background of the proposed system is presented in Section 3. Then, in Section 4, the proposed algorithm is presented. Section 5 presents our results, analysis, and comparisons. Finally, Section 6 concludes the paper.

2 Related Works and Motivation

During the last decade, many steganography related works were proposed in both domains: spatial domain and transform domain. Many methods have been proposed so far for hiding secret information in spatial domain such as; LSB (Least Significant Bit) [4], [5], optimum pixel adjustment process [6], and so on.

The authors in [4] present a scheme which provides two levels of security. It uses RSA Algorithm for encrypting the secret message, then hides it in the four LSBs (Least Significant Bits) of one of the three channels that could be selected through calculating the sum of all pixels in each channel and the one having the maximum value would be the indicator to specify where to embed the secret bits in the other two channels. The experimental results showed that the largest capacity that could be used by the proposed method was 30,116 bytes (240,928 bits) with PSNR (Peak Signal-to-Noise Ratio) value 49.61 dB. However, adopting a combination of cryptography and steganography may increase the security of the system.

Various schemes also have been adopted by the researchers for embedding data in transform domain such as using wavelet transform [7], Discrete Cosine Transform (DCT) [8], Fourier Transform [9], and recently using contourlet transform [10]. The core idea of the last one is embedding the secret message in contourlet coefficients through an iterative embedding procedure to reduce the *stego-image* distortion. Hence, the embedding is done by changing the coefficient values proportional to the regions in which the coefficients reside and hidden data can be retrieved with zero bit error rate. The results showed that using cover selection can embed relatively more bits in a suitable *cover image*. The proposed method is robust against compression but the cost of embedding capacity has been decreased to only 10,000 bits.

After investigating various works, we have found that gaining capacity with visual imperceptibility should be the main objective of any good steganographic scheme. Consequently, we have come up with a reversible steganographic scheme based transform domain. Our adopted transform domain is distinguished from those mentioned in the previous works since it has not been used before in this way in steganographic technique as far as we have investigated in this area. In addition, it is provided with more than one *stego-key*, hence, the proposed method can achieve effective level of security with having reasonable imperceptibility at the same time.

3 Irreducible Polynomial Mathematics

The forward Mix Column Transformation, called Mix Columns, operates on each column individually. Each byte of a column is mapped into a new value that is a function of all four bytes in that column [11]. The results of the Mix Column operation are calculated using $GF(2^8)$ operations. Each element of $GF(2^8)$ is a polynomial of degree 7 with coefficients in $GF(2)$ (or, equivalently Z_2). Thus, the coefficients of each term of the polynomial can take the value 0 or 1. Given that there are 8 terms in an element of $GF(2^8)$, an element can be represented by bit string of length 8, where each bit represents a coefficient. The least significant bit is used to represent the constant of the polynomial, and going from right to left, represents the coefficient of x^i by the bit b_i where b_i is i bits to the left of the least significant bit. For example, the bit string (10101011) represents $(x^7 + x^5 + x^3 + x + 1)$. For convenience, a term x^i is found in the expression if the corresponding coefficient is 1. The term is omitted from the expression if the coefficient is 0. Addition of two elements in $GF(2^8)$ is simply accomplished using eight XOR gates to add corresponding bits. Multiplication of two elements in $GF(2^8)$ requires a bit more work. The multiplication of two elements of Z_2 is simulated with an AND gate. Multiplication in $GF(2^8)$ can then be accomplished by first multiplying each term of the second polynomial with all of the terms of the first polynomial. Each of these products should be added together. If the degree of the new polynomial is greater than 7, then it must be reduced modulo some irreducible polynomial using one of the polynomials which explained in Table 1. In the case of Advanced Encryption Standard (AES), the irreducible polynomial is $x^8 + x^4 + x^3 + x + 1$ [12]. Therefore, multiplication can be performed according to the following rule [11]:

$$x \times f(x) = \begin{cases} (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0) & \text{if } b_7 = 0 \\ (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0) \oplus (00011011) & \text{if } b_7 = 1 \end{cases}$$

In this work, the calculations of Mix Column Transform have been done using $GF(2^3)$ which has not been used before in the literature. Values in $GF(2^3)$ are 3-bits each, spanning the decimal range [0..7]. Multiplication takes place on 3-bit binary values (with modulo 2 addition) and then the result is computed modulo P(x) which can be (1011) = 11 (decimal) or (1101) = 13 (decimal). For example: 5 × 6 = (101) × (110) = (11110) = (011) mod (1011) = 3 (highlighted in Table 1) and 5 × 3 = (101) × (011) = (1111) = (010) mod (1101) = 2 (highlighted in Table 2). Hence, the specific polynomial P(x) provides the modulus for the multiplication results [13].

Table 1. Using Primitive Polynomial (11)

×	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	3	1	7	5
3	3	6	5	7	4	1	2
4	4	3	7	6	2	5	1
5	5	1	4	2	7	3	6
6	6	7	1	5	3	2	4
7	7	5	2	1	6	4	3

Table 2. Using Primitive Polynomial (13)

×	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	5	7	1	3
3	3	6	5	1	2	7	4
4	4	5	1	7	3	2	6
5	5	7	2	3	6	4	1
6	6	1	7	2	4	3	5
7	7	3	4	6	1	5	2

4 Our Proposed Approach

In our work, a distinct kind of transform will be applied on the color images to get new domain for embedding, which is sufficiently secure and can be applied for real-time applications. We present both the embedding and extraction algorithms here.

4.1 Embedding Algorithm

The procedure of embedding is described with the following steps:

Step 1. Dividing the *cover image* into blocks, each block of specified size which can be (3*3), (4*4), (5*5), etc.

Step 2. Selecting some of the blocks for embedding the secret message according to secure key.

Step 3. Pre-processing the specified blocks through taking out the 3 LSBs of from each value and storing it in a new matrix (block).

Step 4. Applying the proposed transform (Mix Column Transform) on each specified block individually.

Step 5. Hiding the secret bits within the matrix after transformation.

Step 6. Applying an inverse transform on the transformed blocks to get back the original blocks.

Step 7. Returning the resulted matrix of 3 LSBs and combing it with original one.

Step 8. Evaluating the proposed method through using the most common measurements that have been used in the literature such as Peak Signal Noise to Ratio (PSNR) and MSSIM for testing the invisibility and the quality of the *stego image*.

4.2 Extraction Algorithm

The proposed method is a blind algorithm so, there is no need for the original *cover image* during the process of extraction. Blind algorithm here refers to the ability of extracting the secret information from the *stego-image* without using the original cover. To recover the secret message, the following steps should be applied:

Step 1. Dividing the *stego-image* into blocks, each block of the same size that has been specified during the embedding.

Step 2. Determining the selected blocks that have been used for embedding the secret message through using the same secure key.

Step 3. Pre-processing the blocks through taking out the 3 LSB's and storing them in a new matrix (block).

Step 4. Applying the proposed transform on each block individually.

Step 5. Extracting the secret bits from the transformed blocks sequentially using secure key.

Step 6. Reconstructing the secret message from the extracted bits.

4.3 The Proposed Transform

In order to apply MCT, it is supposed to have a matrix called transformed matrix which can be generated randomly and should have an inverse. The size of this matrix is variable and can be any. An example could be (3*3) as shown below:

07	01	05
01	06	06
05	06	07

Transformed Matrix

→ $\begin{bmatrix} 011 & 001 & 001 \\ 110 & 011 & 100 \\ 010 & 111 & 001 \end{bmatrix}$

In addition to this matrix, we should have a block matrix taken from a *cover image* with the same size (3*3) which can be referred to as block matrix. Before performing the proposed transform, the block matrix should be pre-processed, then taking the 3 least significant bits from the block matrix and placing in another matrix to get a new one as follows:

179	185	177
182	179	180
178	175	185

$$\begin{bmatrix} 10110011 & 10111001 & 10110001 \\ 10110110 & 10110011 & 10110100 \\ 10110010 & 10101111 & 10111001 \end{bmatrix} \rightarrow \begin{bmatrix} 011 & 001 & 001 \\ 110 & 011 & 100 \\ 010 & 111 & 001 \end{bmatrix}$$

Block Matrix

After that, both matrices have to be converted to polynomials as explained below:

$$\begin{bmatrix} x^2+x+1 & 1 & x^2+1 \\ 1 & x^2+x & x^2+x \\ x^2+1 & x^2+x & x^2+x+1 \end{bmatrix} * \begin{bmatrix} x+1 & 1 & 1 \\ x^2+x & x+1 & x^2 \\ x & x^2+x+1 & 1 \end{bmatrix}$$

Transformed Matrix **Block Matrix**

The proposed transform can be performed via multiplying each row of the transformed matrix with each column of the original values of the block matrix:

$$x^2+x+1 \cdot (x+1) + 1 \cdot (x^2+x) + (x^2+1) \cdot x$$
$$= x^3 + x^2 + x + x^2 + x + 1 + x^2 + x + x^3 + x = x^2 + 1$$

The result is $= x^2 + 1 \rightarrow$ which represents (101) = (5)

The same operation can be done to get the whole values of the resultant matrix which is:

$$\begin{bmatrix} x^2+1 & x & x^2+x \\ x^2+x & x^2 & x \\ x+1 & x^2+x+1 & x^2+x+1 \end{bmatrix} \rightarrow \begin{bmatrix} 101 & 010 & 110 \\ 110 & 100 & 010 \\ 011 & 111 & 111 \end{bmatrix}$$

The largest element appeared in this example is x^2 because the results of the Mix Columns operation are calculated using $GF(2^3)$ operations where, each element of $GF(2^3)$ is a polynomial of the 2nd degree with coefficients in $GF(2)$. Thus, if the result of multiplication leads to get a polynomial with degree larger than 2, then the resultant polynomial should be reduced through dividing it by the irreducible polynomial $(x^3 + x + 1)$ to get the remainder which will be used as a resulted polynomial. Next, the secret message for instance (111) can be embedded in the least significant bit (LSB) of the values of the middle column within the resultant matrix as follows:

$$\begin{bmatrix} 101 & 011 & 110 \\ 110 & 101 & 010 \\ 011 & 111 & 111 \end{bmatrix}$$

On the other hand, to get the original values of the block matrix, the resulting matrix from Mix Column Transform should be multiplied by the inverse matrix:

05	06	02
06	06	04
02	04	02

$$\rightarrow \begin{bmatrix} 101 & 110 & 010 \\ 110 & 110 & 100 \\ 010 & 100 & 010 \end{bmatrix} \rightarrow \begin{bmatrix} x^2+1 & x^2+x & x \\ x^2+x & x^2+x & x^2 \\ x & x^2 & x \end{bmatrix}$$

Inverse Matrix

Again, each row of the inverse matrix will be multiplied by each column of the resulting matrix:

$$\begin{bmatrix} x^2+1 & x^2+x & x \\ x^2+x & x^2+x & x^2 \\ x & x^2 & x \end{bmatrix} * \begin{bmatrix} x^2+1 & x+1 & x^2+x \\ x^2+x & x^2+1 & x \\ x+1 & x^2+x+1 & x^2+x+1 \end{bmatrix}$$

Inverse Matrix **Resulting Matrix**

To get back the first value (03), the first row of the inverse matrix should be multiplied by the first column of the resulted matrix (after transform):

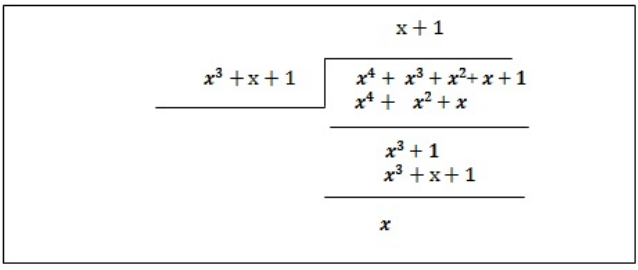
$$(x^2+1)\cdot(x^2+1)+(x^2+x)\cdot(x^2+x)+x\cdot(x+1)$$
$$= x^4+x^2+x^2+1+x^4+x^3+x^3+x^2+x^2+x = x+1$$

The result is $= x + 1$ → which represents (011) = (03)

To get back the second value (0), the first row of the inverse matrix should be multiplied by the second column of the resulted matrix (after transform):

$$(x^2+1)\cdot(x+1)+(x^2+x)\cdot(x^2+1)+x\cdot(x^2+x+1)$$
$$= x^3+x^2+x+1+x^4+x^2+x^3+x+x^3+x^2+x$$
$$= x^4+x^3+x^2+x+1$$

The result is $= x^4+x^3+x^2+x+1$ which has a degree $(4>3)$ so, it should be reduced through dividing it by $(x^3+\mathrm{x}+1)$. This polynomial can be considered as a secret key because it can be changed and it is possible to use either $(x^3+\mathrm{x}+1)$ or (x^3+x^2+1). Therefore, the attacker cannot guess the utilized polynomial in the proposed steganographic algorithm.



Consequently, all other values of the original matrix can be obtained through repeating the same operation.

$$\begin{bmatrix} x+1 & x & 1 \\ x^2+x & x+1 & x^2 \\ x & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 011 & 010 & 001 \\ 110 & 011 & 100 \\ 010 & 001 & 001 \end{bmatrix} \rightarrow \begin{bmatrix} 03 & 02 & 01 \\ 06 & 03 & 04 \\ 02 & 01 & 01 \end{bmatrix}$$

The resulted matrix will again be combined with the block matrix:

$$\begin{bmatrix} 10110011 & 10111010 & 10110001 \\ 10110110 & 10110011 & 10110100 \\ 10110010 & 10101001 & 10111001 \end{bmatrix} \rightarrow \begin{bmatrix} 179 & 186 & 177 \\ 182 & 179 & 180 \\ 178 & 169 & 185 \end{bmatrix}$$

The Block Matrix after applying the transform and embedding the secret message.

Finally, the secret message can be retained through applying the Mix Column Transform on the final resulted matrix for instance:

$$\begin{bmatrix} 07 & 01 & 05 \\ 01 & 06 & 06 \\ 05 & 06 & 07 \end{bmatrix} * \begin{bmatrix} 03 & 02 & 01 \\ 06 & 03 & 04 \\ 02 & 01 & 01 \end{bmatrix}$$

The Transform Matrix **Block Matrix (containing secret message)**

Converting again to polynomials:

$$\begin{bmatrix} x^2 + x + 1 & 1 & x^2 + 1 \\ 1 & x^2 + x & x^2 + x \\ x^2 + 1 & x^2 + x & x^2 + x + 1 \end{bmatrix} * \begin{bmatrix} x + 1 & x & 1 \\ x^2 + x & x + 1 & x^2 \\ x & 1 & 1 \end{bmatrix}$$

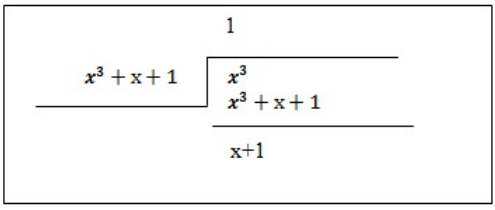
The first value can be got via multiplying the first row of the first matrix with the first column of the second matrix as explained below:

$$(x^2 + x + 1) \cdot (x + 1) + 1 \cdot (x^2 + x) + (x^2 + 1) \cdot x$$
$$= x^3 + x^2 + x^2 + x + x + 1 + x^2 + x + x^3 + x = x^2 + 1$$

The result is $= x^2 + 1$ → which represents (101) = (5)

The second value can be got via multiplying the first row of the first matrix with the second column of the second matrix as explained below:

$$(x^2 + x + 1).x + 1.(x + 1) + (x^2 + 1).1 = x^3 + x^2 + x + x + 1 + x^2 + 1 = x^3$$



The result is $= x + 1$ which is equivalent to (011) = (03)

So, taking the LSB from the resulting value which represents the value of the secret bit, the original value (02) can be obtained.

5 Experimental Results and Discussion

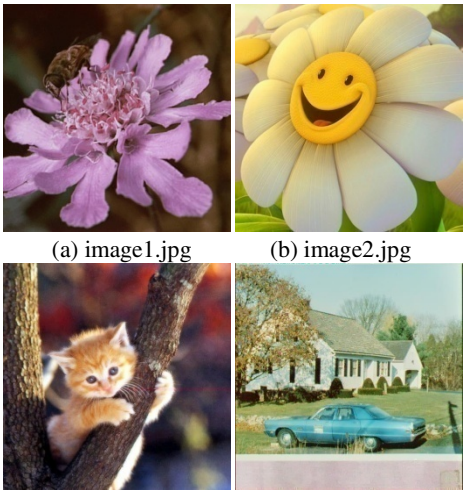
5.1 Experimental Setting

The proposed technique is tested by using sequence of color images of size (512*512) with JPEG formats as shown in Figure 1 (a, b, c, d). The experiments have been conducted using MATLAB [21]. The image quality of the proposed algorithm has been tested using PSNR, which is estimated in decibel (dB) and is defined as:

$$\text{PSNR} = 10 \log \frac{255^2}{\text{MSE}_{avg}}$$

$$\text{MSE} = \frac{1}{\text{hw}} \sum_{i=1}^{\text{h}} \sum_{j=1}^{\text{w}} (x_{ij} - y_{ij})^2$$

where (w and h) denote the width and height of the images respectively. x_{ij} and y_{ij} stand for the value of pixel [i,j] in the original and the processed images, respectively.



(a) image1.jpg (b) image2.jpg

(c) image3.jpg (d) image4.jpg

Fig. 1. Test images for the proposed technique

$$\text{MSE}_{avg} = \frac{MSE_R + MSE_G + MSE_B}{3}$$

where (MSE_R, MSE_G, and MSE_B) are mean square errors in the three channels; Red, Green, and Blue respectively. Table 3 shows the results of applying proposed technique using the mentioned test images [14]. Also Figure 2 and 3 show the output *stego-images*.

Table 3. Results of applying the proposed algorithm on the images of size (512*512)

Color Images of size (512*512)	Payload (Bits)	Block Size	PSNR (dB) of the Stego-image	MSSIM	Embedding Duration Time (seconds)
Image1.jpg	452925	4*4	40.3286	0.9522	100.5894
		8*8	40.3497	0.9529	88.5150
Image2.jpg	452925	4*4	41.2353	0.9515	101.0418
		8*8	40.3330	0.9433	88.2186
Image3.jpg	452925	4*4	40.7893	0.9677	100.6362
		8*8	40.3022	0.9644	88.2186
Image4.jpg	452925	4*4	40.7988	0.9733	99.6066
		8*8	40.3466	0.9714	88.3590

Table 4. Comparison between our proposed method and other related works

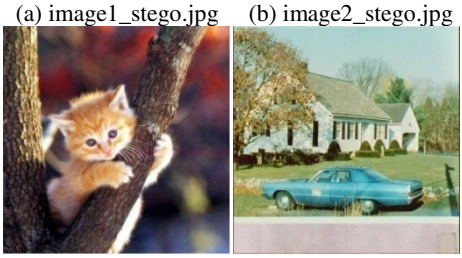
The Steganographic Schemes		The Cover Image	Capacity (Bits)	PSNR of the Stego-image in (dB)	Our Proposed Method		
					PSNR of the Stego-image in (dB)	MSSIM Index	Embedding Duration Time in Seconds
1	**Reference [10]**	Lena .jpg (512*512)	28,001	39.65	47.2571	0.9882	7.6440
2	**Reference [19]**	baboon .bmp (512*512)	162,775	30.02	40.0453	0.9841	36.4106

Another measure for understanding image quality is Mean Structural Similarity (MSSIM) [15] which seems to approximate the perceived visual quality of an image more than PSNR or various other measures. MSSIM index takes values in [0,1] and it increases as the quality increases. We calculate it based on the code in [16] using the default parameters. In case of color images, we extend MSSIM with the simplest way: calculating the MSSIM index of each RGB channel and then, taking the average [17].

5.2 Comparative Analysis

Comparing our proposed scheme with [18] and embedding the same secret message "AB1001CD" within the same *cover image* (baboon.jpg) of size (512*512), we got PSNR=77.3561 while [18] obtained PSNR=72.2156. So, our proposed method beats the scheme used by [18] significantly in terms of imperceptibility through getting higher PSNR. On the other hand, when comparing the proposed scheme with its alternative methods that used gray-scale images in their experiments as presented in [10] and [19], our proposed method exceeds those in terms of invisibility as shown in Table 4 (keeping the capacities same as were used in those schemes).

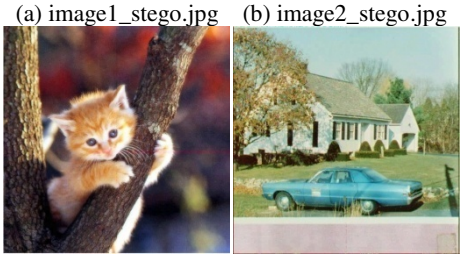
(a) image1_stego.jpg (b) image2_stego.jpg



(c) image3_stego.jpg (d) image4_stego.jpg

Fig. 2. Results of applying the proposed algorithm on the images of size (512*512) using block size (4*4)

(a) image1_stego.jpg (b) image2_stego.jpg



(c) image3_stego.jpg (d) image4_stego.jpg

Fig. 3. Results of applying the proposed algorithm on the images of size (512*512) using block size (8*8)

5.3 Security of the Proposed Transform

According to Kerckhoffs' principle [20], the security of a steganographic system is based on secret key shared between the sender and the receiver called the *stego-key* and, without this key; the attacker should not be able to extract the secret message. In

our proposed method, the secret key was provided in more than one level; firstly the block size is variable and can be any size for instance (3*3), (4*4), etc. Secondly, the transformed matrix is generated randomly and it can be used in our transform if and only if it has inverse. Thirdly, not all the values of the specified block that have been selected for embedding will be used, instead, only 3 LSBs of each value will be taken out and saved separately in another block to be used in our proposed method which has not been used in the literature before. Finally, there is a secret key for selecting the blocks for embedding. That's why the security of our proposed scheme has been significantly increased.

6 Conclusion and Future Work

In this work, we have presented an efficient steganographic method which adopted different style for embedding to increase the security of the system. On the other hand, the capacity of embedding secret message has been maximized without affecting the quality of the *stego-image* as proved by the experiment results for MSSIM measurements which were close to 1. As future work, the robustness of the proposed scheme could be tested against different types of attacks such as the compression to test the efficiency of it and thus, a detailed understanding of the scheme's practicality could be realized.

Acknowledgments. The authors would like to heartily thank the reviewers for their valuable comments that helped improve the paper. This work was supported by NDC Lab, KICT, IIUM.

References

1. Hernandez-Chamorro, A., Espejel-Trujillo, A., Lopez-Hernandez, J., Nakano-Miyatake, M., Perez-Meana, H.: A Methodology of Steganalysis for Images. In: IEEE CONIELECOMP 2009, Cholula, Puebla, Mexico, pp. 102–106 (2009)
2. Li, B., He, J., Huang, J., Shi, Y.Q.: A Survey on Image Steganography and Steganalysis. Journal of Information Hiding and Multimedia Signal Processing 2(2), 142–172 (2011)
3. Lin, C.-C.: An information hiding scheme with minimal image distortion. Computer Standards & Interfaces 33(5), 477–484 (2011)
4. Swain, G., Lenka, S.K.: A Better RGB Channel Based Image Steganography Technique. In: Krishna, P.V., Babu, M.R., Ariwa, E. (eds.) ObCom 2011, Part II. CCIS, vol. 270, pp. 470–478. Springer, Heidelberg (2012)
5. Swain, G., Lenka, S.K.: LSB Array Based Image Steganography Technique by Exploring the Four Least Significant Bits. In: Krishna, P.V., Babu, M.R., Ariwa, E. (eds.) ObCom 2011, Part II. CCIS, vol. 270, pp. 479–488. Springer, Heidelberg (2012)
6. Pandian, N., Thangavel, R.: A Hybrid Embedded Steganography Technique: Optimum Pixel Method and Matrix Embedding. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 1123–1130. ACM (2012)
7. Al-Hunaity, M.F., Najim, S.A., El-Emary, I.M.: Colored Digital Image Watermarking using the Wavelet Technique. American Journal of Applied Sciences 4(9), 658–662 (2007)

8. Liu, Q.: Steganalysis of DCT-Embedding Based Adaptive Steganography and YASS. In: The 13th ACM Multimedia Workshop on Multimedia and Security, pp. 77–85. ACM (2011)
9. Rabie, T.: Digital Image Steganography: An FFT Approach. In: Benlamri, R. (ed.) NDT 2012, Part II. CCIS, vol. 294, pp. 217–230. Springer, Heidelberg (2012)
10. Sajedi, H., Jamzad, M.: Using contourlet transform and cover selection for secure steganography. International Journal of Information Security 9(5), 337–352 (2010)
11. Stallings, W.: Cryptography and Network Security Principles and Practice. Prentice Hall, USA (2006)
12. Li, H., Friggstad, Z.: An Efficient Architecture for the AES Mix Columns Operation. In: Proceeding of ISCAS 2005, Kobe, Japan, pp. 4637–4640 (2005)
13. Addition and Multiplication Tables in Galois Fields GF(2^m), http://www.ee.unb.ca/cgi-bin/tervo/galois3.pl (last accessed May 30, 2013)
14. Yua, Y.-H., Chang, C.-C., Lin, I.-C.: A new steganographic method for color and grayscale image hiding. Computer Vision and Image Understanding 107(3), 183–194 (2007)
15. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
16. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: The SSIM Index for Image Quality Assessment, http://www.cns.nyu.edu/~lcv/ssim/ (last accessed: May 19, 2013)
17. Roussos, A., Maragos, P.: Vector-Valued Image Interpolation by an Anisotropic Diffusion-Projection PDE. In: Sgallari, F., Murli, A., Paragios, N. (eds.) SSVM 2007. LNCS, vol. 4485, pp. 104–115. Springer, Heidelberg (2007)
18. Upreti, K., Verma, K., Sahoo, A.: Variable Bits Secure System for Color Images. In: Proceedings of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 105–107. IEEE (2010)
19. Lee, C.-F., Chen, H.-L., Tso, H.-K.: Embedding capacity raising in reversible data hiding based on prediction of difference expansion. Journal of Systems and Software 83(10), 1864–1872 (2010)
20. Salomon, D.: Coding for Data and Computer Communications, p. 345. Springer (April 12, 2005) ISBN-13: 978-0387212456
21. MATLAB: The Language of Technical Computing, http://www.mathworks.com/products/matlab/ (last accessed May 30, 2013)