DOI: 10.21917/ijivp.2013.0081

DIGITAL COLOR IMAGE ENCRYPTION BASED ON INVERTIBLE MATRIX WITH SECRET SHARING

W.R. Sam Emmanuel[1] and C. Edward Jaya Singh[2]
Department of Computer Science, Nesamony Memorial Christian College, India
E-mail: [1]sam_emmanuel@yahoo.com and [2]cmedwardsingh@yahoo.com

Abstract
This paper explores the new approach to implement image encryption in digital color images. The self invertible matrix created from the original image is used as keys for the RGB to YCbCr transform and the secret sharing operations. The encryption process carried out by the four steps: pixel permutation, creating RGB matrix, RGB to YCbCr transform and the secret sharing. The quality of the encrypted images are tested with visual inspection and evaluated with different quality measures. The performance of the proposed method is also evaluated by various testing methods.

Keywords:
Self-Invertible Matrix, Pixel Permutation, Secret Sharing

1. INTRODUCTION

The security of digital images in the computer field plays key role in this internet age. The encryption techniques[6] try to convert an image to another image which is hard to understand; to keep the image confidential between users. It is essential that nobody could get to know the content without a key for decryption. Special and reliable security in storage and transmission of digital images is needed in many applications, such as medical imaging systems, military image communications and confidential video conferences, etc. In order to fulfill such a task, many types of encryption schemes have been proposed for secure transmission of images. It is always necessary to develop more and more secure image encryption techniques, because of the intruder's domination in the network.

Many image encryption techniques are used only for the grayscale images[7][8]. The natural images can be transferred insecure form after doing encryption[10]. The encryption in color images has lot of complexities[9][11]. The color image encryption based on secret sharing proposed with different computation matrices in various levels[3].

This paper presents the color image encryption based on the self invertible matrix operation. The quality of the encrypted images are tested with visual inspection and evaluated with different quality measures. The performance of the proposed method also evaluated. Rest of the paper is organized as follows.

The section 2 elaborates the pixel permutation, Self-invertible matrix generation, secret sharing and the transform between RGB and YCbCr. Section 3 explores the proposed method. The section 4 list-out the experimental results. The results obtained from the simulations are discussed in this section. Finally, the concluding remarks are given in section 5.

2. THE FUNDAMENTAL OPERATIONS

The digital gray scale images can be secured using various transforms and pixel operations. In the color image encryption scheme, the three different frames are maintained. Each frame will be processed separately by various stages and the pixel values are directly input for secret sharing. The working process involved in the proposed method are explained here.

2.1 SELF-INVERTIBLE MATRIX

As the color image decryption requires inverse of the matrix, there arises a problem whether the inverse of the key matrix does exist or not during decryption. If the matrix is not invertible, then the encrypted text cannot be decrypted. In order to overcome this problem, the self-invertible matrix[1][4][5] is used in the colour image encryption. In the self-invertible matrix generation method, the matrix used for the encryption is itself self-invertible. So, at the time of decryption, it is not necessary to find the inverse of the key matrix. Moreover, this method eliminates the computational complexity involved in finding the inverse of the matrix during decryption. A is called self-invertible matrix if $A = A - 1$.

2.2 PIXEL PERMUTATION

The image encryption schemes are mainly consisting of image pixel permutation[2] stage and pixel diffusion stage. Confusion refers to making the relationship between the key and the encrypted image as complex as possible. The confusion stage is composed of pixel permutation. Here the pixel permutation is defined by,

$$PP[i, j] = OI[1 + (77i + 3) \bmod 256, 1 + (19j + 3) \bmod 256] \quad (1)$$

where, $OI[i, j]$ represents the $(i, j)^{th}$ pixel of the original image and $PP[i, j]$ denotes the $(i, j)^{th}$ pixel of the permutation resulted image. This method uses two constant values to improve the pixel scrambling of the image. In the decryption process, the pixel permutation is replaced by inverse pixel permutation. The inverse pixel permutation method is described by,

$$OI[i, j] = PP[1 + (133i - 4) \bmod 256, 1 + (27j - 4) \bmod 256] \quad (2)$$

2.3 RGB TO YCBCR COLOR SPACE TRANSFORM

The RGB to YCbCr color space transform[3] is the process to convert the RGB space image to YCbCr space image. If the of RGB space image components are R, G and B then the converted components of YCbCr space image are Y, Cb and Cr respectively. The YCbCr system is a popular luminance-chrominance space. The main advantage of luminance-

chrominance system is that some of the chrominance information can be discarded without causing noticeable artifacts. The RGB to YCbCr space mapping is defined as,

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{255} SIK_1 \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \bmod 256 \quad (3)$$

where, *Y* stands for the luminance component, *Cb* and *Cr* represent the blue and red chrominance components.

For the YCbCr to RGB color space transform, the YCbCr space image is converted in the form of RGB space image. The conversion processes is given as,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{255} SIK_1^{-1} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \bmod 256 \quad (4)$$

2.4 SECRET SHARING

The secret sharing is the process in which the YCbCr space image will be encrypted. The single step cannot scramble the pixel values evenly. If the operations repeated for several times then the pixel values may distributed evenly. The encryption operation is given as,

$$SS_i = SIK_2 SS_{i-1} \bmod 256 \quad (5)$$

where, $i = 2,3\ldots n$, $SS_1 = SIK_2 IT$, $SS_1 = \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$ and $IT = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$.

For the inverse secret sharing process the encrypted image is converted in the form of YCbCr space image. The decryption process is given as

$$ISS_{i-1} = SIK_2^{-1} ISS_i \bmod 256 \quad (6)$$

where, $i = n,..,3,2$, $IIT = SIK_2^{-1} ISS_1$, $ISS_i = \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$ and $IIT_i = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$

3. PROPOSED METHOD

This section explained the key generation, encryption and decryption steps of the proposed method. The original image, $m \times n$ size has three color channels, each having the pixel value range 0 to 255. The size of every color channels is in the size of $m \times n$ array, which represents the colors red (*R*), green (*G*) and blue (*B*).

3.1 KEY GENERATION

Choose any one pixel from the original image. Separate the *R*, *G* and *B* values of the pixel and generate a 2 × 2 order matrix A_{22} by assuming the fourth element as zero. Apply self invertible matrix generation with this 2 × 2 order matrix. This 2 × 2 order matrix will produce 4 × 4 order matrix, *A*. The matrix *A* will be in the following form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where, A_{11}, A_{12}, A_{21} and A_{22} are matrices of order 2 × 2.

The different combinations of the elements in the 4 × 4 order matrix will generate 3 × 3 order matrices. Choose any two 3 × 3 order matrix, which can be used as self invertible keys SIK_1 and SIK_2. The SIK_1 is used in the RGB to YCbCr image transform and the SIK_2 is used for secret sharing. The various stages of key generation shown in the Fig.1.

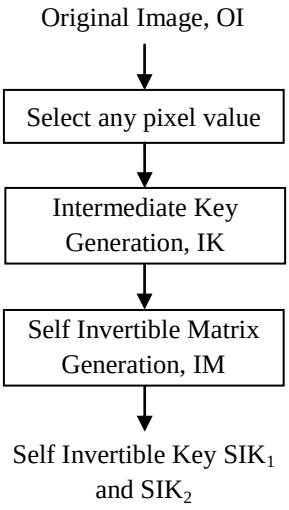

Fig.1. Key Generation Module

3.2 ENCRYPTION

The original color image, *OI* has three different frames with the variations of the colors *R*, *G* and *B*. The different colors are stored separately in the three different frames. Apply the pixel permutation (Eq.(1)) in each frame independently. This process scatters the values of each colors in different places. It will create the confusion in the pixel storing location. The pixel permutation applied in the three different places separately makes more confusion of the pixel location.

The permutation applied bit planes are stored separately. Collect the *R*, *G* and *B* values of each pixel and store it in the form of 3 × 1 order matrix. This matrix help the transformation from RGB to YCbCr. The RGB to YCbCr transformation is done with the help of self invertible key SIK_1. The conversion carried out by the Eq.(3).

The transformed result produced *Y*, *Cb* and *Cr* values for every pixels. This YCbCr values converted as an encrypted image by the secret sharing operation. The secret sharing step is done with the values YCbCr and self invertible key SIK_2. The secret sharing (*SS*) process can be repeated number of times to get the information more secure. In every i^{th} iteration $(i\text{-}1)^{th}$ value is used for calculating the *SSi* value. The Eq.(5) is used for getting the encrypted pixels. This will produce the encrypted image *EI*. The various modules and the levels of execution are displayed in Fig.2.

3.3 DECRYPTION

The decryption process will produce the original image. The decryption is the reverse process of encryption. The encrypted image is converted in the YCbCr form by applying the inverse secret sharing (*ISS*) stage. The inverse of the self invertible key

(SIK_2^{-1}) is used with the encrypted image and the process done in number of times by Eq.(6).

The YCbCr form image is converted to RGB form by the inverse image transform (*IIT*). The inverse of self-invertible key SIK_1^{-1} is used with the YCbCr form will produce the RGB matrix. The conversion is done by the Eq.(4), this generates the number of RGB matrix corresponding to the YCbCr matrix. Generate the pixels from the RGB matrix by creating three different plances for *R*, *G* and *B*.

The inverse pixel permutation applied with these three planes (*R*, *G* and *B*) separately. Combined form of these three planes will produce the original color image. The different levels of operations are visualized in Fig.2.

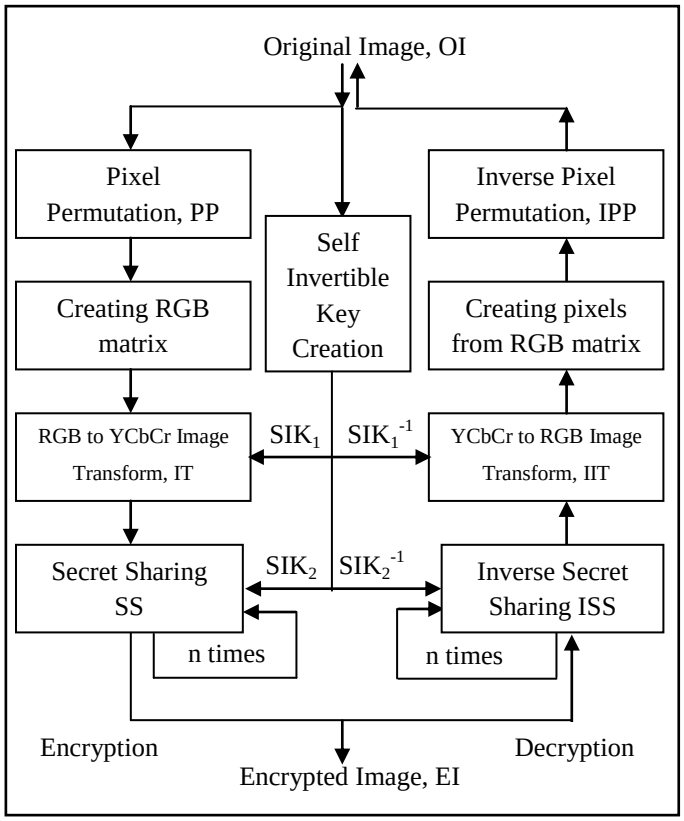

Fig.2.The Encryption and Decryption Process

4. RESULTS AND DISCUSSIONS

The proposed method is tested by three different color images. The images considered here with the same width and height (ie. 256 × 256). The color image 'Lena' with the size of 21.0KB, Fig.3(a) produced the encrypted image Fig.3(d), it gives the decrypted image Fig.3(g). The color image 'Peppers' with the size of 25.5KB Fig.3(b) produced the encrypted image Fig.3(e), it gives the decrypted image Fig.3(h). The color image 'Girl' with the size of 14.2KB, Fig.3(c) produced the encrypted image Fig.3(f), it gives the decrypted image Fig.3(i).

It is best to maintain the Encryption Algorithm healthy from cryptanalytic, statistical and brute force attacks. Generally some analyses which supports security through Histogram Analysis, Correlation of Two Adjacent Pixels, Key Sensitivity Analysis and Number of pixels change rate (NPCR). Here, the proposed method is analyzed using the three images (Lena, Peppers and Girl) by the subsequent Security Analysis techniques. The time complexity also calculated for this new approach.

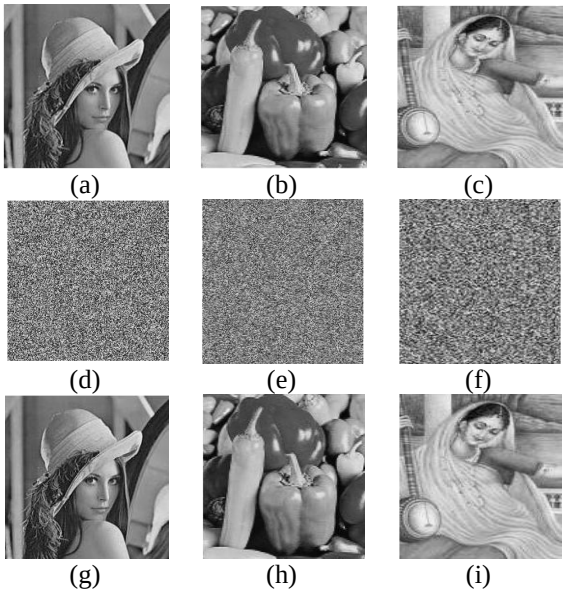

Fig.3. The original images (a-c), the encrypted images (d-f) and the decrypted images (g-i)

4.1 HISTOGRAM ANALYSIS

Histogram analysis is used to demonstrate the superior confusion and diffusion properties of the encrypted image. The color variations in Red, Green and Blue of the original image and the encrypted image obtained by the scheme are displayed in the form of histograms.

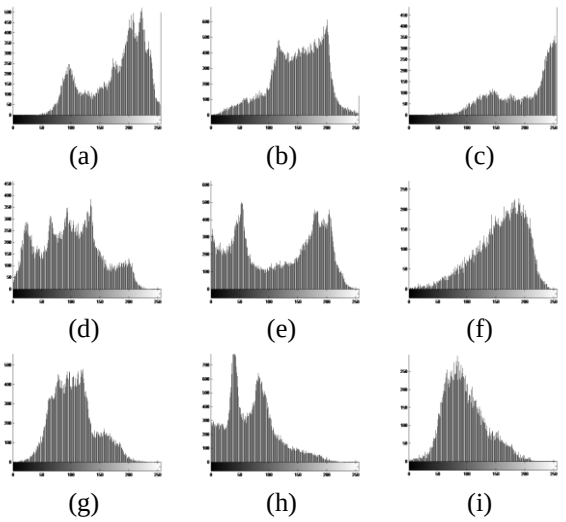

Fig.4. RGB channel histograms of original images

The RGB channels of the original images Fig.3(a), Fig.3(b) and Fig.3(c) are shown in Fig.4(a-c), Fig.4(d-f) and Fig.4(g-i) respectively. The RGB channels of the encrypted images Fig.3(d), Fig.3(e) and Fig.3(f) are shown in Fig.5(a-c), Fig.5(d-f) and Fig.5(g-i) respectively.

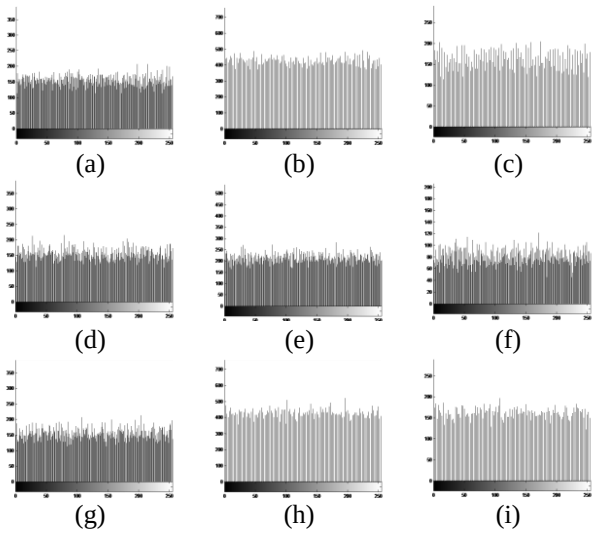

Fig.5. RGB channel histograms of encrypted images

The histograms of the original and encrypted images are compared, and it is observed that the histogram of the encrypted image is fairly uniform and is significantly different from that of the original image. Hence, the encrypted images transmitted do not provide any distrust to the attacker, so it is concluded that this method can strongly resist histogram based attacks.

4.2 CORRELATION OF TWO ADJACENT PIXELS

The correlation between two adjacent pixels in plain image and encrypted image can be tested by using the following equations which also used to calculate the correlation coefficients in horizontal, vertical and diagonal directions.

The correlation coefficient between two variable x and y usually denoted by r_{xy} and is defined as

$$r_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y} \tag{7}$$

where,

The covariance $Cov(x,y) = \frac{1}{n}\left[\sum (x-\bar{x})(y-\bar{y})\right]$,

Standard deviation of x is $\sigma_x = \sqrt{\frac{1}{n}\left[\sum (x-\bar{x})^2\right]}$ and

Mean $\bar{x} = \frac{1}{n}\sum x$

where, x and y are the R or G or B value of two adjacent pixels in the image. At random choose pairs of adjacent (horizontal, vertical and diagonal direction) pixels from original image and encrypted image, and compute the correlation coefficients.

In the Table.1, two adjacent pixels in the original image are highly associated and the correlation coefficient is nearly equal to 1, whereas in the encrypted image, it is approximately equal to 0, so they are highly uncorrelated.

4.3 COMPLEXITY

Apart from the security consideration, running speed of the algorithm is also an important aspect for a good encryption algorithm. The encryption/decryption rates of several colored images are measured for different size by using the proposed image encryption scheme. The time analysis has been done on Pentium Dual Core CPU with 1.2 GHz RAM computer. The average encryption and decryption time taken by the algorithm for different sized images are shown in the Table.2.

4.4 KEY SENSITIVITY ANALYSIS

An ideal image encryption procedure should be sensitive with respect to the secret key. The change of a single bit in the secret key should produce a completely different encrypted image, which means that the encrypted image cannot be decrypted correctly although there is only a slight difference between encryption and decryption keys. This guarantees the security of the proposed method against brute-force attacks to some extent.

It is not easy to compare the encrypted images by simply observing these images. The correlation between the corresponding pixels of the three encrypted images is compared for getting good result. Use the Eq.(7) for this calculation. Here the x and y are the values of corresponding pixels in the two encrypted images to be compared. Table.3, gives the results of the correlation coefficients between the corresponding pixels of the three encrypted images. It is clear from the table that no correlation exists among three encrypted images even though these have been produced by using slightly different secret keys. Key sensitivity analysis shows that changing one bit in encryption key will result in a completely different encrypted image.

Table.1. Correlation Coefficient Analysis

Images	Direction	Original Image			Encrypted Image		
		Red	Green	Blue	Red	Green	Blue
Lenna	Diagonal	0.8117	0.8342	0.7530	0.0570	0.0474	0.1983
	Vertical	0.9198	0.9461	0.9078	0.0415	-0.224	-0.052
	Horizontal	0.9016	0.9045	0.8281	-0.0448	0.0587	0.0020
Peppers	Diagonal	0.9570	0.9608	0.9213	-0.0038	0.0891	0.0084
	Vertical	0.9582	0.9653	0.9093	-0.0064	0.1074	-0.128
	Horizontal	0.8980	0.9703	0.9260	-0.0335	-0.100	0.2220
Girls	Diagonal	0.9336	0.9008	0.8448	0.0199	-0.087	-0.023
	Vertical	0.9589	0.9258	0.8432	0.1127	0.0243	0.0480
	Horizontal	0.8708	0.9383	0.8267	0.1242	-0.035	0.0523

Table.2. The Time Complexity Analysis

Image	Image Size	Encryption Time (in Seconds)	Decryption Time(in Seconds)
Lenna	21.0KB	0.9680	0.9426
Peppers	25.5 KB	1.4530	1.2391
Girl	14.2 KB	0.4380	0.4276

Table.3. Key Sensitivity Analysis

Image	Key1	Key2	Correlation Between encrypted images using Key1 and Key2		
			Red	Green	Blue
Lenna	2181371200	2181371201	-0.1295	0.0289	-0.2791
Peppers	16794390	16794391	-0.2475	-0.1955	-0.1924
Girl	1391301210	1391301211	-0.0554	-0.1373	-0.1823

4.5 NPCR AND UACI

The number of pixels change rate (*NPCR*) is measured to see the influence of changing a single pixel in the original image on the encrypted image by the proposed algorithm. The *NPCR* measure the percentage of different pixel numbers between the two images. We take two encrypted images, C_1 and C_2, whose corresponding original images have only one-pixel difference. We define a two-dimensional array D, having the same size as the image C_1/C_2. The $D(i, j)$ is determined from $C_1(i, j)$ and $C_2(i, j)$. If $C_1(i, j) = C_2(i, j)$ then $D(i, j) = 1$ otherwise $D(i, j) = 0$. The *NPCR* is defined by the Eq.(8).

$$NCPR = \frac{\sum D(i, j)}{wh} * 100\% \quad (8)$$

where, w and h are the width and height of encrypted image. The *NPCR* obtained for a large number of images by using our encryption scheme and found it to be over 98% showing thereby that the encryption scheme is very sensitive with respect to small changes in the plaintext. This is shown in Table.4.

Table.4. NPCR Measures

Image	NPCR (in %)			
	Red	Green	Blue	For the Pixel
Lenna	97.6382	97.4774	97.6558	97.5905
Peppers	97.7741	97.5228	97.9399	97.7456
Girl	98.9628	98.6651	99.0589	98.8956

5. CONCLUSION

In this paper, the color image encryption done based on the invertible matrix with secret sharing. The key values are generated from the invertible matrix of the original image. It is observed that the qualities of the proposed method are better than the Hao Luo's color image encryption method. It also state that instead of using the constant values in the color transform and secret sharing stages, it is better to use the variables for getting good quality encryption. The complicated key values may produce harder security.

REFERENCES

[1] Shanmugam P and Loganathan C, "Involutory Matrix In Visual Cryptography", *International Journal of Research and Reviews in Applied Sciences*, Vol. 6, No. 4, pp. 424-428, 2011.

[2] Shatheesh Sam I, Devaraj P and Bhuvaneswaran R. S, "Block Cipher Scheme for Image Cryptosystem Using Alternative Chaotic Maps", *European Journal of Scientific Research*, Vol. 51, No. 2, pp. 232-240, 2011.

[3] Hao Luo, Fa-Xin Yn, Hui Li and Zheng-Liang Huang, "Color Image Encryption Based on Secret Sharing and Iterations", *Information Technology Journal*, pp. 1-7, 2010

[4] Bibhudendra Acharya, Saroj Kumar Panigrahy, Sarat Kumar Patra and Ganapati Panda, "Image Encryption Using Advanced Hill Cipher Algorithm", *International Journal of Recent Trends in Engineering*, Vol. 1, No. 1, pp. 663-667, 2009.

[5] Bibhudendra Acharya, Girija Sankar Rath, Sarat Kumar Patra and Saroj Kumar Panigrahy, "Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm", *International Journal of Security*, Vol. 1, No. 1, pp. 14-21, 2007.

[6] Stallings W, "*Cryptography and Network Security*", Prentice Hall, 2005.

[7] Lee W, Chen T and Chieh Lee C, "Improvement of an encryption scheme for binary images", *Pakistan Journal of Information and Technology*, Vol. 2, No. 2, pp. 191-200, 2003.

[8] Mitra A, Subba Rao Y V and Prasanna S. R. M, "A new image encryption approach using combinational permutation techniques", *International Journal of Electrical and Computer Engineering*, Vol. 1, No. 2, pp. 127-131, 2006

[9] Koga H and Yamamoto H, "Proposal of a lattice-based visual secret sharing scheme for color and gray-scale images", *Institute of Electronics, Information and Communication Engineers Transaction on Fundamentals*, Vol. E81-A, No. 6, pp. 1262–1269, 1998.

[10] Nakajima M and Yamaguchi Y, "Extended visual cryptography for natural images", *Journal of WSCG*, Vol. 2, pp. 303–310, 2002.

[11] Sam Emmanuel W.R, "Multilevel Information Hiding Using Image Encryption and Image Steganography" *International Journal of Cryptography and Security*, Vol. 1, No. 2, pp. 15-20, 2008.