Secure Reversible Data in Encrypted JPEG Image Compression with Proposed Modulo Pixel Component Selection Method

Anukriti Saxena[1] Prof. Ravi Mohan[2]
[1,2]SRIT, Jabalpur, India

Abstract— Data hiding is the art of hiding data for various purposes such as--- to maintain private data, secure confidential data. Well known technique is the Steganography; Steganography has evolved into a digital strategy of hiding a file in some form of multimedia, such as an image, an audio file or even a video file. This paper presents a new Steganography technique, In this paper, we propose a new high-capacity reversible data hiding method for JPEG-compressed images. This method is based on modifying the quantization table and quantized discrete cosine transformation (DCT) coefficients. Some elements of the quantization table are divided by an integer while the corresponding quantized DCT coefficients are multiplied by the same integer and added by an adjustment value to make space for embedding the data. By analyzing the effect of each single quantized DCT coefficient on the image quality, an embedding sequence is chosen in order to help control the increase of file size after hiding the data meanwhile the PSNR value between the original uncompressed image and stego JPEG image is high. Experimental results show that the proposed method achieves both high capacity and high image quality.

Keywords: DWT: Discrete Wave Transform, SVD: Singular Value Decomposition, PSNR: Peak Signal to Noise Ratio, MSE: Mean Square Error

I. INTRODUCTION

In the proposed scheme, a cover JPEG image is first decoded to get the quantization table and quantized DCT coefficients, then some entries of the quantization table are divided by an integer and corresponding quantized DCT coefficients are multiplied by the same integer and added by an adjustment value to make space for embedding the data. After extracting secret bits from the stego image, the original JPEG image can be recovered at the same time. The proposed algorithm can be divided into two phases: the data hiding phase and the extracting and restoring phase which are illustrated in the following sections. To reduce the distortion caused by embedding and control the increase of the file size, the selection of embedding positions should be discussed first.

The selection of embedding positions For the purpose of making the influence caused by data hiding as small as possible, the selection of embedding positions should be carefully considered. As Huang et al. (2000) said, traditional data hiding techniques tended to choose the mid-frequency DCT coefficients in the DCT transform domain. However, when considering the quantization stage, things may be different. Thus, the earlier techniques might affect the researchers' choice on embedding positions in the JPEG compressed domain. Xuan et al. (2007) did some simple experimental investigation to decide their optimum parameters. To test the effect of every single quantized DCT coefficient on image quality in much more detail, theoretic analysis and experimental investigation are done in this paper. Here the peak signal-to-noise ratio (PSNR) is adopted to evaluate the impact:

$$\mathrm{PSNR} = 10\log10\left(\frac{255^2}{MSE}\right)$$

Where MSE (mean squared error) for an M×N grayscale image is defined as:

$$\mathrm{MSE} = \frac{1}{\mathrm{MxN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f'(x,y) - f(x,y))^2$$

Where f_(x, y) and f(x, y) are the pixel values of the distorted image and the original image respectively.

II. METHODOLOGY

Figure 1 below shows the proposed work flow diagram the overall procedure can be understand as by following steps:-

1) Step 1: input image acquisition can be any taken and image but proposed work use MATLAB standard images of lena, jet, Barbara, baboon and peppers.
2) Step 2: isolate the Red, Green and Blue frames as the image processing are suitable at 2D matrix only and all the reaming mathematics performed on Red, green, blue individually and at the last all this frame gets concatenate again.
3) Step 3: separate 8x8 block out of 2D frame as DCT is applicable only on 8x8 block

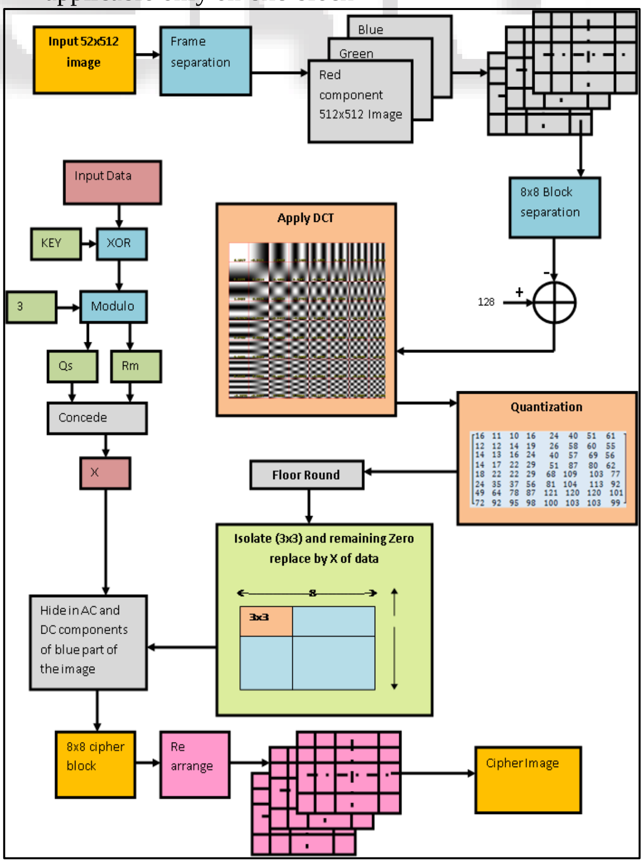

Fig. 1: block diagram of proposed work

4) Step 4: subtract the 8x8 block by 128 as cosine signals varies between -1 to 1, out frame block will vary between -128 to 128.
5) Step 5: apply DCT on the output block from step 4.
6) Step 6: perform Quantization by the quantization table provided by JPEG2 standard.
7) Step 7: round the output of quantization at its floor value.
8) Step 8: output 8x8 block will have few values in its 3x3 frame (9 pixels) rest of the 55 pixels will remains zero, the row zero elements are known as AC components and columns zero will be known as DC components.
9) Step 9: input the data and convert the data into ASCII.
10) Step 10: perform XOR between ASCII data and input KEY.
11) Step 11: compute modulo 3 operation on the data until each pixels gets convert into set of modulii
12) Step 12: concatenate the all modulo output
13) Step 13: hide the modulo components into AC and DC elements of the output frame from step 8.
14) Step 14: rearrange the 8x8 block and perform all step 4 to step8 operation into new 8x8 block
15) Step 15: concatenate red, green and blue frame.

III. ALGORITHM

Image acquisition: Let an image is x (i, j, k)

Where 'i' is the row number, 'j' is column number, 'k' is frame (k=1 for red, k=2 for green & k=3 for blue) and x is the intensity of pixel at (i, j, k).

i= 1: M

j=1: N

k=1: 3

Hence size of image is MxNx3

Sub-block development:

$c_k(p,q) = x(\,m:m+7, n:n+7, 1:3\,)$

Where m= 1 to M with interval of 8 as m=1, 9, 17,…. M

Where n= 1 to N with interval of 8 as n=1, 9, 17, ….N

And k=1, 2, 3,……… (MxN/64)

p=1,2,….8

q=1,2,….8

Subtraction by 128

$d_k(p,q) = \{128 - c_k(p,q)\}$

Cosine coefficient matrix and image

	s=1	2	3	4	5	6	7	8	r=
cosθ (r,s)	6.1917	−0.3411	1.2418	0.1492	0.1583	0.2742	−0.0724	0.0561	1
	0.2205	0.0214	0.4503	0.3947	−0.7846	−0.4391	0.1001	−0.2554	2
	1.0423	0.2214	−1.0017	−0.2720	0.0789	−0.1952	0.2801	0.4713	3
	−0.2340	−0.0392	−0.2617	−0.2866	0.6351	0.3501	−0.1433	0.3550	4
	0.2750	0.0226	0.1229	0.2183	−0.2583	−0.0742	−0.2042	−0.5906	5
	0.0653	0.0428	−0.4721	−0.2905	0.4745	0.2875	−0.0284	−0.1311	6
	0.3169	0.0541	−0.1033	−0.0225	−0.0056	0.1017	−0.1650	−0.1500	7
	−0.2970	−0.0627	0.1960	0.0644	−0.1136	−0.1031	0.1887	0.1444	8

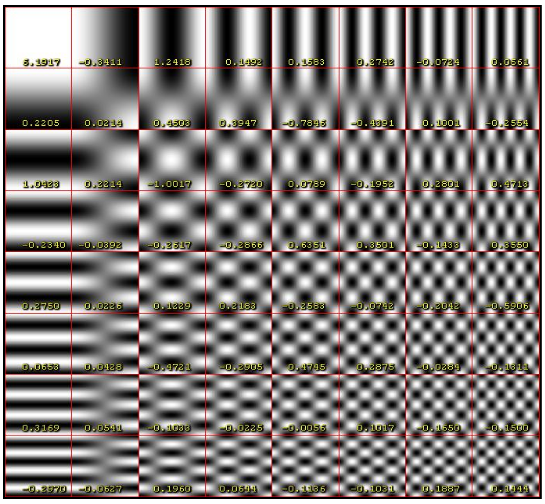

Fig. 2: DCT cosine components

r=1, 2,….8 is the row numb er of cosine coefficient image

s=1, 2,….8 is the column number of cosine coefficient image

Combination of each part of cosine coeffecient images to get sub image d_k

$A_{pq}(1,1) * \cos\theta(1,1) = d_k(p,q)$

$A_{pq}(1,2) * \cos\theta(1,2) = d_k(p,q)$

………………

…………

$A_{pq}(1,8) * \cos\theta(1,8) = d_k(p,q)$

$A_{pq}(2,1) * \cos\theta(2,1) = d_k(p,q)$

$A_{pq}(2,2) * \cos\theta(2,2) = d_k(p,q)$

………………

…………

$A_{pq}(8,8) * \cos\theta(8,8) = d_k(p,q)$

$$e_k(r,s) = \sum_{p=1}^{8}\sum_{q=1}^{8} A_{pq}(r,s)$$

Quantization

$$Q_{rs} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 22 & 29 & 68 & 109 & 103 & 77 \\ 24 & 35 & 37 & 56 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 121 & 120 & 120 & 101 \\ 72 & 92 & 95 & 98 & 100 & 103 & 103 & 99 \end{bmatrix}$$

Q_{rs} given above is the JPEG quantization matrix

$$f_k(r,s) = \frac{e_k(r,s)}{Q_{rs}}$$

$$g_k(r,s) = \text{floor}|f_k(r,s)|$$

$g_k(r,s)$ is the quantized Image sub-block

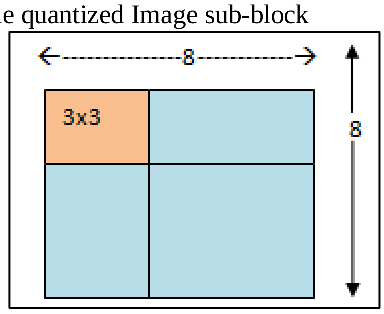

Fig. 3: 3x3 separation from zero value AC and DC components

A. Zig-Zag Arrangement

$h_k(1,ct) = g_k(a,b)$ where $b = 1,2,\ldots.i$ and $a = (i+1) - b$] for $i = 1,3,5,7$

ct=ct+1;

$h_k(1,ct) = g_k(a,b)$ where $a = 1,2,\ldots.i$ and $b = (i+1) - a$] for $i = 2,4,6,8$

ct=ct+1;

$h_k(1,ct) = g_k(a,b)$ where $a = 8,7,\ldots.i$ and $b = i, i+1 \ldots.a$] for $i = 2,4,6,8$

ct=ct+1;

$h_k(1,ct) = g_k(a,b)$ where $b = 8,7,\ldots.i$ and $a = i, i+1 \ldots.b$] for $i = 3,5,7$

ct=ct+1;

where ct=r*s

Apply Run length Coding

Each non zero encoded as triple

[(r, s),c]

Where r number of zeros before current value'

s is the size of number in bits

c is the actual value

$$r_p = r_p + 1 \text{ when } (h_k(1,i) \text{ xor } h_k(1,i+1) == 0) \text{ else } p = p + 1$$

$$s_p = \text{count}\{(s)_2\}$$

$$c_p = h_k(1,i)$$

$$\text{for } i = 1,2 \ldots.ct$$

Now can be consider as compressed and coded data stream for the original jpeg image sub-block c_k (8x8) is $[r_p, s_p, c_p]$ and $g_k(t,u)$ is the compressed image sub-block where t=1,2,3 and u = 1,2,3

$$c_k(8x8) \rightarrow g_k(3x3)$$

Let K is the KEY for encryption, then

$$l_k = g_k \text{ xor } K$$

Let D is the data which needs to be hide

$$D1 = (D)_2$$

As l_k is a 3x3 sub-image hence 9 pixels

$(y_k(1,1))_2 = (l_k(1,1))_2$ OR (0000000&D1(1,1)

$(y_k(1,2))_2 = (l_k(1,2))_2$ OR (0000000&D1(1,2)

$(y_k(1,3))_2 = (l_k(1,3))_2$ OR (0000000&D1(1,3)

$(y_k(2,1))_2 = (l_k(2,1))_2$ OR (0000000&D1(1,4)

$(y_k(2,2))_2 = (l_k(2,2))_2$ OR (0000000&D1(1,5)

$(y_k(2,3))_2 = (l_k(2,3))_2$ OR (0000000&D1(1,6)

$(y_k(3,1))_2 = (l_k(3,1))_2$ OR (0000000&D1(1,7)

$(y_k(3,2))_2 = (l_k(3,2))_2$ OR (0000000&D1(1,8)

$(y_k(3,3))_2 = (l_k(3,3))_2$ OR (0000000&D1(1,9)

y_k will have the data hidden at its LSB and which will be not effected by JPEG compression

IV. RESULT

Figure 4 below shows the original and cipher image of Lena develop using proposed work.

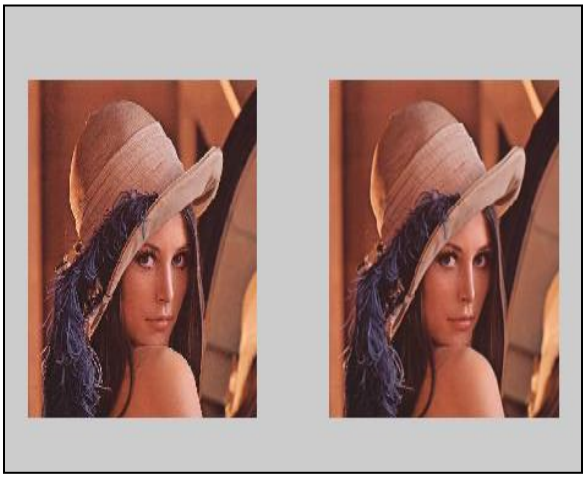

Fig. 4: input and output image

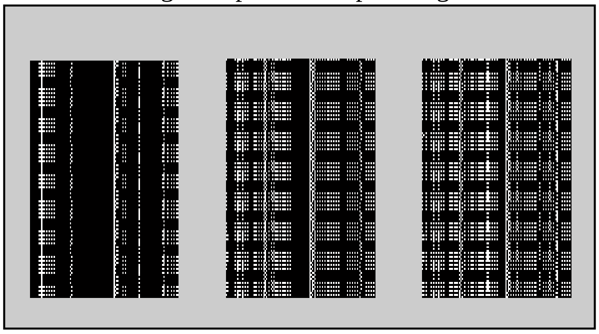

Fig. 5: JPEG pixels after run length coding

Figure 5 above shows the binary image of red green and vlue frame after run length coding applied on the lena image.

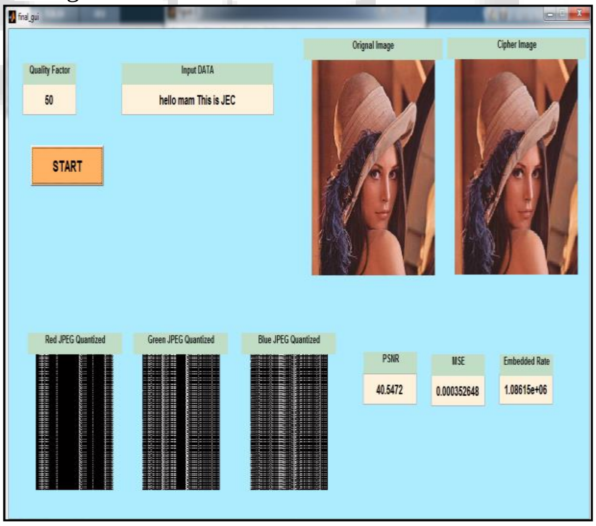

Fig. 6: Proposed work GUI

Figure 6 above shoes the proposed work GUI for input and output observation and parameter input and output observation.

QF=50			
	embedding rate R	PSNR	MSE
Lena	10.67	40.54	0.0003526
barbara	10.07	40.89	0.000325
Pepper	11.22	39.1339	0.000488
Jet	10.67	41.27	0.000698
Baboon	10.06	41.68	0.000671

Table 1: Obtain results parameters

QF=50				
	Proposed work		Xiaozhu Xie et al [1]	
	embedding rate R	PSNR	embedding rate R	PSNR
Lena	10.67	40.54	6.43	38.09
barbara	10.07	40.89	5.44	35.76
Pepper	11.22	39.13	6.38	37.22
Jet	10.67	41.27	6.35	38.34
Baboon	10.06	41.68	4.42	32.87
Average	10.53	40.90	5.80	36.46

Table 2: Comparative results

Table 1 above shows the obtain results for the MATLAB standard image of lena, baboon, Barbara, Jet and papper, all input images of 512x512 size and the data hidden is of 1 Kb.

Table 2 above shows the comparative results between Xiaozhu Xie et al [1] work and proposed work it can be observe that proposed work PSNR is higher then [1] work, also the data embedding rate is high.

V. Conclusion

In this paper, a high capacity reversible JPEG-to-JPEG data hiding scheme is proposed. Through lowering certain quantization table entries and lifting corresponding quantized DCT coefficients, space is made for embedding data. Using the proposed embedding strategy, our scheme can achieve high embedding capacity and keep the distortion introduced by embedding very low, meanwhile the original cover JPEG image can be restored after the secret data are extracted. Experiments results demonstrate that the proposed scheme maintains the image quality of a stego JPEG image when the embedding capacity is high. Besides, the file size after embedding with not too huge data is acceptable. Compared with Chang et al.'s method and Xuan et al.'s method, the proposed method is superior in terms of the image quality, hiding capacity and file size. The proposed scheme is very practical for image files stored and transmitted in the JPEG format.

Reference

[1] Xiaozhu Xie, Chin-Chen Chang, Reversible Data Hiding in Encrypted Images Using Reformed JPEG Compression, Natural Science Foundation of P. R. China under Grant 61503316, Natural Science Foundation of Fujian Province under Grant 2016J01326., IEEE 2017
[2] Ziqiang Cheng, Kee-Young Yoo, A Reversible JPEG-to-JPEG Data Hiding Technique, 2009 Fourth International Conference on Innovative Computing, Information and Control, 978-0-7695-3873-0/2009 IEEE
[3] Abdelhamid Awad Attaby, Mona F.M. Mursi Ahmed, Abdelwahab K. Alsammak, Data hiding inside JPEG images with high resistance to steganalysis using a novel technique: DCT-M3, Ain Shams Engineering Journal, ScienceDirect, 10.1016/j.asej.2017.02.003, 2017
[4] Lu´ıs F. R. Lucas, Nuno M. M. Rodrigues, Luis A. da Silva Cruz and S´ergio M. M. de Faria, Lossless Compression of Medical Images Using 3D Predictors, DOI 10.1109/TMI.2017.2714640, IEEE, Transactions on Medical Imaging
[5] Paul T. Chiou, Yu Sun, and G. S. Young A Complexity Analysis of the JPEG Image Compression Algorithm, 978-1-5386-3007-5/17/2017 IEEE
[6] Almohammad, A., Ghinea, G., Hierons, R.M., 2009. JPEG steganography: a performance evaluation of quantization tables. In: Proceedings of IEEE International Conference on Advanced Information Networking and Applications, Bradford, United Kingdom, pp. 471–478.
[7] Cheng, Z., Yoo, K.Y., 2009. A reversible JPEG-to-JPEG data hiding technique. In: Proceedings of 2009 Fourth International Conference on Innovative Computing, Information and Control, Kaohsiung, Taiwan, pp. 635–638.
[8] Fridrich, J., Goljan, M., Du, R., 2001. Invertible authentication watermark for JPEG images. In: Proceedings of IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, pp. 223–227.
[9] Fridrich, J., Goljan, M., Du, R., 2002. Lossless data embedding for all image formats. In: Proceedings of SPIE, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV, vol. 4675, San Jose, CA, pp. 572–583.
[10] Huang, J., Shi, Y.Q., Shi, Y., 2000. Embedding image watermarks in DC components. IEEE Transactions on Circuits and Systems for Video Technology 10 (6), 974–979.
[11] Iwata, M., Miyake, K., Shiozaki, A., 2004. Digital steganography utilizing features of JPEG images. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E87-A (4), 929–936.
[12] Lin, P., Chan, C., 2010. Invertible secret image sharing with steganography. Pattern Recognition Letters 31 (13), 1887–1893.
[13] Sakai, H., Kuribayashi, M., Morii, M., 2008. Adaptive reversible data hiding for JPEG images. In: Proceedings of International Symposium on Information Theory and its Applications, Auckland, New Zealand, pp. 1–6.
[14] Upham, D., 1997. JPEG-JSteg. http://www.funet.fi/pub/crypt/steganography/jpegjsteg-v4.diff.gz
[15] Wallace, G.K., 1991. The JPEG still picture compression standard. Communications of the ACM 34 (4), 30–44.
[16] Westfeld, A., 2001. F5-a steganographic algorithm: high capacity despite better steganalysis. In: Proceedings of the 4th International Workshop on Information Hiding, Pittsburgh, PA, USA, pp. 289–302.
[17] Xuan, G.R., Shi, Y.Q., Ni, Z.C., Chai, P.Q., Cui, X., Tong, X.F., 2007. Reversible data hiding for JPEG images based on histogram pairs. In: Proceedings of International Conference on Image Analysis and Recognition, Montreal, Canada, pp. 715–727.
[18] Zhang, X., Wang, S., Qian, Z., Feng, G., 2010. Reversible fragile watermarking for locating tampered

blocks in JPEG images. Signal Processing 90 (12), 3026–3036.

[19] Chang, C.C., Chen, T.S., Chung, L.Z., 2002. A steganographic method based upon JPEG and quantization table modification. Information Sciences 141 (1–2), 123–138.

[20] Chang, C.C., Lin, C.C., Tseng, C.S., Tai, W.L., 2007. Reversible hiding in DCT-based compressed images. Information Sciences 177 (13), 2768–2786

