INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DOUBLE COMPRESSION BASED REVERSIBLE DATA HIDING IN ENCRYPTED IMAGE

R. Sathesh Raaj[1], Mr. R. MohanRaj[2], Mr. P. Parthasarathy[3]

[1]Assistant Professor/ECE PSNA College of Engineering and Technology, Dindigul.
[2]Assistant Professor/ECE, PSNA College of Engineering and Technology, Dindigul
[3]Assistant Professor/ECE, PSNA College of Engineering and Technology, Dindigul
Email: [1]satheshraaj@gmail.com[1], mohaedu14@gmail.com[2], sarathy1990@gmail.com[3]

Abstract: - The Protecting privacy for exchanging information through the media has been a topic researched by many people. Up to now, cryptography has always had its ultimate role in protecting the secrecy between the sender and the intended receiver. However, nowadays steganography techniques are used increasingly besides cryptography to add more protective layer to the hidden data. In this letter, we show that the quality factor in a JPEG image can be an embedding space, and we discuss the ability of embedding a message to a JPEG image by managing JPEG quantization tables (QTs). In combination with some permutation algorithms, this scheme can be used as a tool for secret communication. The proposed method can achieve satisfactory decoded results with this straightforward JPEG double compression strategy.

Index terms: - Data hiding, JPEG, quality factor.

I. INTRODUCTION

ALONG with the demand for speed and integrity while exchanging information over the Internet, there is always the need for secrecy. Many works has focused on how to protect private information from being attacked and/or identified. Besides cryptography, steganography is also increasingly used for secure communication [1], [2]. Different from cryptography, the main goal of data hiding is to conceal the hidden data by the carrier media, so that the hidden data is transferred without drawing suspicions.

The hiding algorithms are to maintain the natural appearance of the cover media and to keep uninvolved people from even thinking the information exists. To hide information inside an image, there are several available domains where steganography algorithms exploit such as spatial domain or DCT domain [3]. Among various types of images, JPEG format is a commonly used standard of lossy compression for photographic images. JPEG images can typically gain 30:1 compression ratio with little perceptible loss in image quality. Another advantage of JPEG standard is that the degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality.

In this work, a secret communication scheme is proposed, in which the data is doubly protected by both encryption stage and hiding stage. The message to be embedded is first processed by applying some encryption techniques. Herein, the permutation algorithm that requires a pair of numbers as a key is employed to permute the original message. After the encryption stage, the scrambled message is then embedded into a JPEG image by managing different quantization tables. The final result is a JPEG image containing some certain regions with different image quality. The recipient performs reverse steps to extract the information: first extract the pattern of the scrambled message, and then use the key which was shared previously to decipher the message.

II. PROPOSED HIDDEN INFORMATION EMBEDDING/EXTRACTING SCHEME

A. JPEG Image Compression Process Revisited:

The standard JPEG is mostly employed to deal with color images in RGB format. The first step is to convert the image from RGB space into luminance/chrominance spaces Y, Cb and Cr. Color space conversion step is followed by the subsampling step, where typically the chrominance channels (Cb and Cr) are subsampled with the rate equal to half of the rate of the Y channel. By applying the specific quantization tables (QTs), the high frequency components of the transformed image is Truncated and upper left components is separated.
A larger coefficients set in a quantization table leads to a higher compression rate, but also reduces the image quality, and vice versa. Different camera models use different QTs for the same quality. For example, the two QTs in the following figure, are different, but they provide the same approximate quality factor (PSNR) of 80.

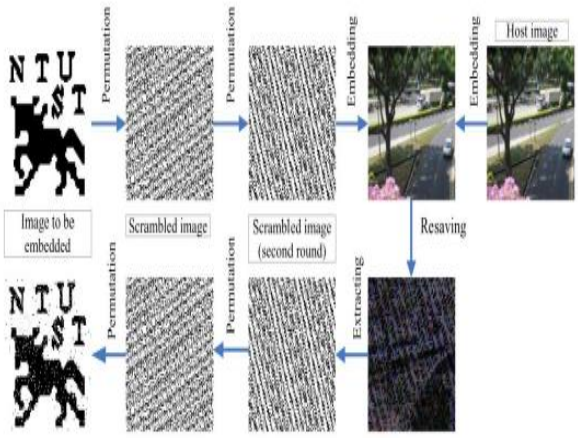
6	4	4	6	10	16	20	24	6	4	3	6	9	15	19	22
5	5	6	8	10	23	24	22	4	4	5	7	9	21	22	30
6	5	6	10	16	23	28	22	5	4	6	9	15	21	25	21
6	7	9	12	20	35	32	28	5	6	8	10	19	32	30	23
7	9	15	22	27	44	41	31	6	8	13	21	25	40	38	28
10	14	22	26	32	42	45	37	9	13	20	24	30	39	42	34
20	26	31	35	41	48	48	40	18	24	29	32	38	45	45	37
29	37	38	39	45	40	41	40	27	34	35	36	42	37	38	37

Fig 1: QT1 and QT2 with same quality factor.

Consider the case when a set of coefficients is quantized by an amount q1, i.e., a block of size 8 8 contains DCT coefficients $c_{i,j}$ from c_{11} to c_{88} maps to a QT of the same size. Each DCT coefficient is quantized by the corresponding amount in QT and then rounded to the nearest integer.

$$C\hat{}_{i,j} = \text{round}(c_{i,j} / q_{i,j})$$

If that DCT set is recalculated ($c_{i,j} = C\hat{}_{i,j} * q_{i,j}$) and then the set is quantized a second time by an amount q2.



SYSTEM MODEL:

Fig 2: Embedding and extracting message

III. ENCRYPTION STAGE:

The image to be embedded is first passed through an encryption stage. Here we simply permute the pixels in the hidden message. In fact, any encryption algorithm can be applied in this stage, provided that the algorithm itself and the key are strong enough**.** The image to be embedded is first passed through an encryption stage. Here we simply apply an auto-morphism algorithm [7], [8] to permute the pixels in the hidden message. In fact, any encryption algorithm can be applied in this stage, provided that the algorithm itself and the key are strong enough. An image, after applying some modulo operation, becomes a random pattern. For example, after applying this operator with parameter on the original image 66 times, we have the scrambled image like the image next to the original one; is the parameter for changing the divisor in this modulo operation and is considered the key for decryption. To recover the original image, one has to know the pair . In this case, to decode we apply the same operator with and (with this specific setup, applying the operation 192 times yields the original image). To be even more secure, in practice this step can be repeated with several rounds and different sets and can be applied. Hence, the key that the recipient needs to reconstruct hidden image is the set of those and values.

IV. EMBEDDING PROCEDURE:

Let the Host Image is 'H' which is the Binary or Color Image of size P*Q. Where P=Width of the Host Image H,Q=Height of the Host Image H.To keep the shape of the secret image unchanged, the aspect ratio of the secret image should be identical to that of the host image.

Aspect Ratio: P/Q=M/N

For Eg: H Size=1024*1024 Pixels, where S Size=128*128 Pixels.
The above Images only satisfy the condition of Aspect Ratio.
Step 1:
Divide the original host image into blocks of size P/M*Q/N.
For 1024*1024 'H' and 128*128 'S', the Host Image is spitted in to blocks of size 8*8.
Hence the Host Image contains 16 8*8 Sub-blocks.
Step 2:
If S(m,n)=0 then H(i,j) is compressed with the Quantization Table Q2.

If S(m,n)=1 then H(i,j) is compressed with the Quantization Table Q1.
Hence the Each Sub-block in Host Image is quantized either Q1 or Q2, which leads to Double Compression.
Step 3:
Convert the Embedded Image in to JPEG Format.
In Existing Compression method, the Host Image is quantized with Q1 and again quantized with Q2 which leads to low quality.
In our Proposed work, the quality should be as high as possible, in that way the pixel values will not be quantized and rounded again with another large QTs. After the embedding process, we save the embedded image in JPEG format, but the quality should be as high as possible, in that way the pixel values will not be quantized and rounded again with another large QTs. Quality for saving the final image after embedding data was chosen 100 or 95 in the experiments for the above reason. In decoding the recipient first detects the regions with different quality factors to extract the pattern of the message embedded to the host image (the scrambled pattern). As the pattern is extracted, the recipient can rescramble the pattern with the key which was shared in advance with the sender to reconstruct the hidden message.

V. DECODING MODULE:

Step 1: Resave the embedded image with lower quality factors (Q2).

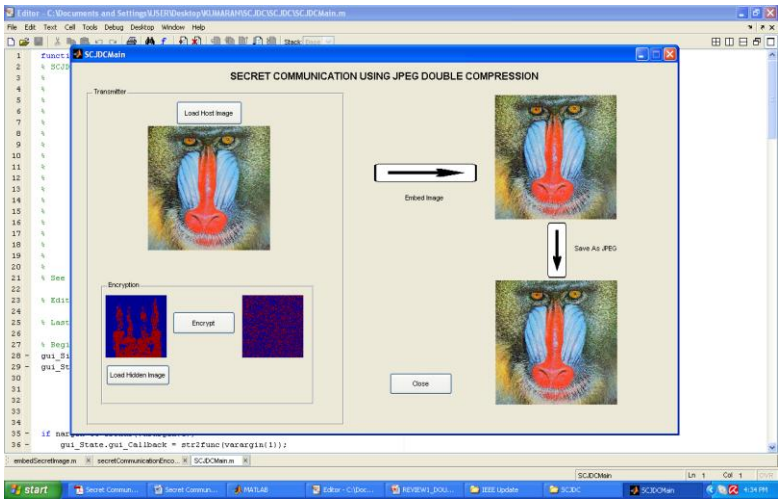
Step 2: Subtract the resaved version to the original embedded image to obtain the difference image.

Step 3: Divide the difference images into blocks of size 8* 8

Step 4: The greatest and least sum values in the difference image are denoted as and Select the difference image which has the greatest difference as the best candidate. Set the threshold.

Step 5: If the sum of the pixel value in each block is smaller than the threshold then decode a black pixel, otherwise decode a white pixel. In Matlab, the function imwrite allows us to write JPEG image from a matrix that represents the uncompressed image, with specified quality factor defined by user.
The default QTs at different quality factors of other software such as Photoshop were also examined to test the ability to extract the embedded information with blind guess of QTs.

VI. SIMULATION RESULTS:



VII. REFERENCES:

[1] K. Raja and C. Chowdary, "A secure image steganography using LSB, DCT and compression techniques on raw images," in *3rd Int. Conf. Intelligent Sensing and Information Processing*, 2005, pp. 170–176.

[2] Z. Ni and Y. Shi, "Robust lossless image data hiding designed for semi-fragile image authentication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 497–509, 2008.

[3] G. Gul and F. Kurugollu, "A novel universal steganalyser design: "LogSv"," in *IEEE Int. Conf. Image Processing (ICIP 2009)*, Cairo, Egypt, 2009.

[4] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. Inform. Forensics Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.

[5] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York: Springer, 1993.

[6] [Online].Available: http://www.impulseadventure.com/photo/ jpegquantization.html

[7] G. Voyatzis and I. Pitas, "Applications of toral auto-morphisms in image watermarking," in *Int. Conf. Image Processing, Proceedings*, 1996, vol. 1, pp. 237–240.

[8] H. K. Tso *et al.*, "A lossless secret image sharing method," in *8th Int. Conf. Intelligent Systems Designs and Application*, 2008, pp. 616–619.

[9] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 1, pp. 81–84, Jan. 2002

[10] S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machine," in *SPIE Symp. Electronic Imaging*, 2004 [Online].Available: http://www.cs.dartmouth.edu/farid/research/