iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30301

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

High Capacity Reversible Data Hiding in Encrypted JPEG Images by Quadtree Compression and Prediction Error

Aiswarya Mohan[1], Prof. Helen K J[2]
[1, 2]*Department of Computer Science and Engineering, Government Engineering College, Thrissur, Kerala, India.*

Abstract: ***The existing techniques of reversible data hiding in encrypted JPEG images vacates space for data hiding after encrypting the image. As the entropy of encrypted images is higher, there is a chance for information loss as well as the computational complexity of image reconstruction will be higher. The proposed method therefore finds space for data hiding before encrypting the image using the method of quadtree compression. The simplicity of quadtree data structure reduces computational complexity. The image is encrypted using RC4 encryption algorithm. The vacant spaces obtained after quadtree compression is used for data hiding with the help of a data hiding key. At the receiver side, data extraction and image recovery are made separable such that the data can be extracted without knowing the image contents. The original pixel restoration is done using prediction error method. The experimental results demonstrate that the proposed method is able to achieve better data hiding capacity and recovered image quality within an acceptable computational complexity range.***
Keywords: Reversible data hiding, Quadtree, Prediction error.

I. INTRODUCTION

Reversible data hiding (RDH) is the process by which secret data can be concealed from viewers by hiding it in a cover medium and can be recovered losslessly from the deformed cover medium [1]. This has been an important research area for many years. The different types of cover medium used for RDH are images, audio, video etc. Among these, digital images are mostly used as the cover medium. Cryptography and data hiding are two methods used to protect the confidentiality of digital images. The first is to transform the image into a noisy image to prevent unauthorized access, while the other is to embed the secret message in a cover image imperceptibly. Reversible data hiding makes it possible to recover the cover image perfectly after extracting the secret data. Recently, security is a major concern due to the development of cloud computing and because of this reversible data hiding is done in enciphered images. Therefore, RDHEI caught the attention of many researchers. It can be used in many areas, such as medical imaging, forensics and military applications. RDHEI can be done by two methods namely vacating room after encryption (VRAE) and reserving room before encryption (RRBE).

Nowadays, JPEG (Joint Photographic Experts Group) is the mostly used digital image format by cameras and other photo-video devices [2]. Along with the wide spread usage in military and medical areas, JPEG images are used in our day to day lives. It is the mostly used image format for photograph storage and their transmission. RDH in a JPEG image is also used in authenticating images and managing archives. In multimedia archives, the provider of the image does not want any changes in the image data, even if the changes are not detectable to the users and also the storage requirements of both the host image and sealed image will be expensive. It is sometimes undesirable for the owner of the image to change the image slightly to authenticate the image. In such cases, RDH is the perfect choice as it can detect fake areas and recover the contents of the original image.

The existing data hiding methods are mainly using images which are not compressed as test images and have achieved good results over the last few years as they have better recovered image quality and hiding capability. Generally, these methods cannot be used for JPEG images. This is because, data hiding mainly look for redundant data in the original image and as JPEG image is already compressed, the amount of redundant data will be smaller. Also in JPEG images, already quantization is done and because of that, changes in the discrete cosine transform (DCT) values will cause more distortions. Along with the image quality for a JPEG image, the storage requirements are also to be checked as it can be more after inserting the secret data. All these reasons make reversible data hiding harder in JPEG images than in uncompressed images.

Mainly, four different types of methods are developed for RDH in JPEG images [2]. The first method is based on lossless compression. The second method is done by modifying the JPEG quantization tables. The third method is performed by doing some modifications to the Huffman table and the last method is done by performing modifications to the quantized DCT coefficients.

The proposed framework uses the first approach of lossless compression as the computational complexity of other methods are higher. The existing reversible data hiding methods in encrypted JPEG images vacates space for hiding the data after image encryption. Limitations of this method are low data hiding capacity and conditional reversibility. Since the entropy is maximized for encrypted images, it is difficult to find more space for additional data using compression, pixel correlation etc. in the encrypted domain. Also, error free extraction of data and reversibility of cover image may not be possible at high embedding rates. To overcome these drawbacks, the proposed framework uses the method of reserving room before encryption.

II. LITERATURE REVIEW

This section briefly discusses some of the recent studies in reversible data hiding in JPEG images along with their drawbacks.

In paper [3], a padding strategy is used to embed an adjustable amount of information in the JPEG images. The algorithm modifies only a small number of zero-valued quantized DCT coefficients to embed the message. The message to be embedded is represented in ternary instead of in binary. The image size is increased to slight extent because of modifying zero valued coefficients as it affects the efficiency of run length encoding done during JPEG compression.

In paper [2], a method of histogram shifting is used for reversibly hiding data in JPEG images. The zero coefficients remain unchanged and only coefficients with values 1 and -1 are expanded to carry the message bits. A novel block selection strategy is used where flatter blocks are selected for data hiding to reduce the invalid shifting of outer coefficients. But this block ordering method is not very reasonable when there are many blocks with the same number of zero AC coefficients. This is because these blocks may differ in the quantity of embeddable and shiftable.

In paper [4], an iterative recovery algorithm is used for hiding data in JPEG images. The method is separable as anyone who has the embedding key can extract the additional message from the marked encrypted bit stream without revealing the original content of the JPEG image. Here the average complexity of recovery is higher as a lot of computation is required. Also, the VRAE approach used will result in information loss.

In paper [5], a JPEG encryption algorithm is provided to encipher a JPEG image to a smaller size and keep the format compliant to JPEG decoders. During data hiding, a combined embedding algorithm including two stages is proposed, the Huffman code mapping and the ordered histogram shifting. The embedding procedure is reversible. Additional messages can be extracted from the marked encrypted JPEG bit stream and the original encrypted bit stream can be recovered losslessly. The authorized user obtains the original JPEG bit stream by a direct decryption. VRAE approach reduces recovered image quality. The hiding capacity is more but the computational complexity of embedding is high.

All the existing methods hide data in the encrypted JPEG images by vacating space after encryption which results in information loss while data extraction and image recovery processes. To overcome the drawbacks of the existing methods, the proposed framework reserves space for data hiding before encryption. The quadtree data structure is used for image compression and the space for data hiding is obtained after image compression. Secret data is embedded into the computed space using prediction error technique. Data extraction and image recovery are separable. The main objectives of the proposed method are to reduce the computational complexity, improve the data hiding capacity and recovered image quality.

III. PROPOSED METHOD

The proposed system mainly consists of four steps: image compression using a quadtree, image encryption, data hiding in the encrypted image, data extraction and image recovery. The architecture of the proposed system is shown in Fig. 1.

A. Image Compression Using a Quadtree

The quadtree is an efficient data structure to represent an image. The quadtree is used for image compression as well as for image decomposition. It is based on “divide and conquer” strategy. The quadtree is a hierarchical data structure, in which each node has 4 children. The quadtree is based on the concept of recursive decomposition. Quadtree can be classified into region quadtree, point quadtree etc. By using the quadtree entire image is subdivided into four equal-sized quadrants. The quadrants are represented as North West (NW), North East (NE), South West (SW) and South East (SE). One of the features of the quadtree is that space is decomposed into adaptable cells. Every cell in the quadtree has a maximum capacity. After reaching the maximum capacity, the cells are split again. Consider a quadtree of depth ‘n’. This quadtree represents an image of size $2^n * 2^n$. Each pixel value in the quadtree should be a ‘0’ or ‘1’. The root node of the quadtree is used to represent the entire image. The image is subdivided again until all the pixels in the same region are either 0’s or 1’s. A block of pixels is represented by the leaf node indicating 0’s or 1’s. The quadtrees can be used in applications of collision detection, image representation etc.

For quadtree compression, the image is converted into a binary representation. Quadtrees are constructed from the binary representation of the image. The lowest level of the quadtree would contain N nodes equivalent to the number of pixels in the image [6]. The next level would contain N/4 nodes. Thus, the total number of nodes necessary can be found by N + N/4 + N/16 + . . . +1. Therefore, the entire memory required for the quadtree manipulation is O(N). Here the image can be compressed using the pruning technique. In this, leaves with values close to that of their parents are removed. This is continuously done until no further leaves can be removed. The lowest level is then taken to form the image, using only leaf nodes. Thus, after the compression, the vacant spaces are obtained and the pixel positions of the pruned pixels are noted by performing a bottom-up scanning after quadtree compression. These pixel positions are later used for data hiding.

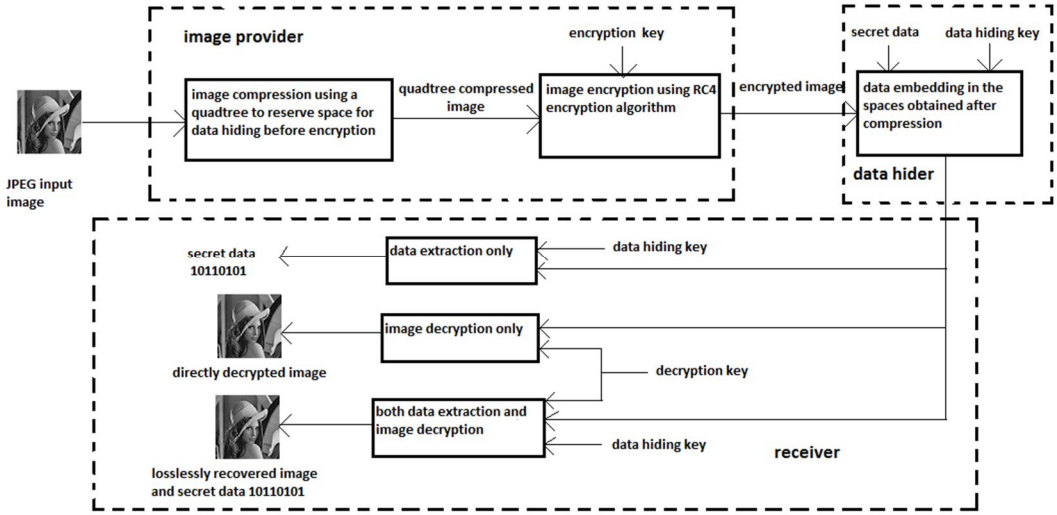


Fig. 1 Architecture of the proposed work

B. Image Encryption

Image obtained after quadtree compression is encrypted using RC4 encryption algorithm. RC4 was designed by Ron Rivest which is officially termed as "Rivest Cipher 4" [7]. RC4 is a symmetric key cipher (stream cipher) algorithm. This means that the same key is used for both encryption and decryption. At a time, RC4 algorithm will encrypt one byte. A key input is actually a pseudo random bit generator that generates an 8-bit number which cannot be predicted if the input key is unknown. The output produced by the generator is known as a key stream and it is combined with the plain text stream cipher one byte at a time using the exclusive OR operation. The advantage of using RC4 is that it is extremely fast and uses small amount of RAM. Also RC4 algorithm is cryptographically very strong and easy to implement.

C. Data Hiding in the Encrypted Image

In the data hiding phase, the data hider embeds the additional data into the encrypted image. Secret data are embedded in the vacant positions obtained after the quadtree compression. Data hiding in these positions are done with the help of data hiding key. The border pixels of the encrypted image are not allowed for data hiding. If a pixel is chosen for data hiding, then its four neighbouring pixels are not allowed to modify. Firstly, the pixels whose pixel values close to their parent pixel are used for data hiding. After the data are embedded into these positions, again the pixels are randomly chosen for data hiding.

Using the data hiding key, data hider selects l pixels from the encrypted image. Let the length of the secret data to be embedded be l. Let the secret data to be embedded be d_1, d_2,.., d_l. The selected l pixels can be represented as P_1, P_2,.., P_l. The additional data is embedded into the t^{th} bit of selected pixel. Here, the Most Significant Bit (MSB) is used for data hiding. Data embedding is done using the equation given below. Let j^{th} pixel is selected to embed the j^{th} bit of secret data.

$$P_j' = P_j - p * 2^{(t-1)} + d_j * 2^{(t-1)}$$

P_j' is the modified encrypted pixel containing the secret data bit and p is computed as,

$$p = \left\lfloor \frac{P_j}{2^{(t-1)}} \right\rfloor mod2$$

D. Data Extraction and Image Recovery

Data extraction and image recovery includes three cases depending upon the role of the recipient [8]. If the recipient has only the data hiding key, then he can extract the embedded secret data without knowing the contents of the original image. If the recipient has only the decryption key, then he can decrypt the image by RC4 decryption and get the directly decrypted image. This image will be slightly distorted. If both the data hiding key and the decryption key is available with the recipient, then he can extract the embedded secret data as well as recover the original image without any loss. Finally, a high-quality image will be obtained. The modified pixel restoration is done using the prediction error mechanism.

1) *Receiver has Only the Data Hiding Key:* Here, the receiver is the data owner. He first selects the l pixels with the help of data hiding key. Let P_1', P_2', .., P_j',..,P_l' be the retrieved pixels. Then the l embedded secret data can be extracted using the below equation, where $1 \leq j \leq l$.

$$d_j = \left\lfloor \frac{P_j'}{2^{(t-1)}} \right\rfloor mod2$$

2) *Receiver has Only the Decryption Key:* By using the decryption key, the receiver generates pseudo-random bits. For each 8 bit of pixel, 8 bits of pseudo-random bits are generated. Each bit of pseudo-random bits can be represented as $r_k(i, j)$. By using these pseudo-random bits each pixel value can be decrypted to form a directly decrypted image. But the directly decrypted images are prone to some distortion.

3) *Receiver has both the Decryption and Data Hiding Keys:* If the receiver has both the keys, then he can recover the original image without any error as well as he can extract the secret data. With the help of encryption, the receiver will generate a pseudo-random bit represented as $r_k(i, j)$. These bits are used for decrypting the encrypted pixels. By using the data hiding key, the receiver can extract the embedded bits in the selected pixels. The embedded bits are obtained by extracting the t^{th} bit of selected pixel which is the MSB bit of that selected pixel. Only the t^{th} bits of the selected pixels will differ from the original image, in the case of a directly decrypted image. These pixels can be restored by using the prediction error mechanism. Let $E_1, E_2, \ldots E_l$ be the selected l pixels in the directly decrypted image. The value of a pixel can be predicted by using four of its neighbouring pixels. Let $E_{i,j}$ be a pixel value in the encrypted image. The value can be estimated using four of its neighbouring pixels, $E_{i,j-1}$, $E_{i,j+1}$, $E_{i-1,j}$, $E_{i+1,j}$ through two orthogonal directions 0^0 and 90^0. In this proposed method, two prediction values γ_0 and γ_{90} are computed.

$$\gamma_0 = \frac{E_{(i,j-1)} + E_{(i,j+1)}}{2}$$

$$\gamma_{90} = \frac{E_{(i-1,j)} + E_{(i+1,j)}}{2}$$

Then select an optimal pair of weights w_0 and w_{90} to give a good estimate value.

$$E_{i,j}^* = \gamma_0 w_0 + \gamma_{90} w_{90}$$

$$w_0 = \frac{\sigma_{90}}{\sigma_{90} + \sigma_0}$$

$$w_{90} = 1 - w_0$$

Where,

$$\sigma_0 = \frac{1}{3}\sum_{k=1}^{3}(\lambda_0(k) - u)^2$$

$$\sigma_{90} = \frac{1}{3}\sum_{k=1}^{3}(\lambda_{90}(k) - u)^2$$

And,

$$\lambda_0 = \{E(i, j-1), E(i, j+1), \gamma_0\}$$

$$\lambda_{90} = \{E(i-1, j), E(i+1, j), \gamma_{90}\}$$

$$u = \frac{1}{4}\{E(i, j-1), E(i, j+1), E(i-1, j) + E(i+1, j)\}$$

The estimated value E_j be E_j^*, where $1 \leq j \leq l$. Two possible values of E_j is obtained by setting the t^{th} bit as 0 and 1 represented by,

$$E_j^0 = E_j - p * 2^{(t-1)} + 1 * 2^{(t-1)}$$

$$E_j^1 = E_j - p * 2^{(t-1)}$$

Where,

$$p = \left\lfloor \frac{E_j}{2^{(t-1)}} \right\rfloor mod2$$

Two prediction errors ER_j^0 and ER_j^1 are calculated by,

$$ER_j^0 = |E_j^0 - E_j^*|$$

$$ER_j^1 = |E_j^1 - E_j^*|$$

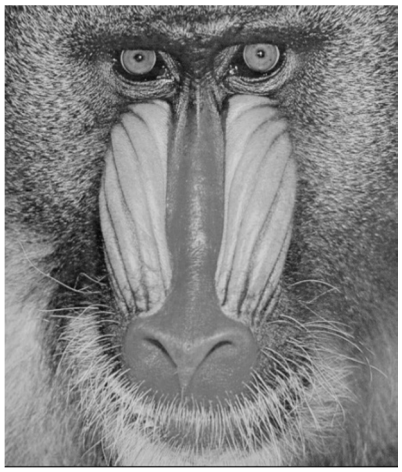
The output original pixel can be determined by the equation,

$$E_{jout} = \begin{cases} E_j^0, & if\ ER_j^0 \leq ER_j^1 \\ E_j^1, & if\ ER_j^0 \geq ER_j^1 \end{cases}$$

IV. EXPERIMENTS AND RESULTS

The proposed method is conducted with an experimental environment of MATLAB R2018a under windows 8. JPEG 8-bit grayscale images of size 512 X 512 are taken as test images as shown in Fig. 2.

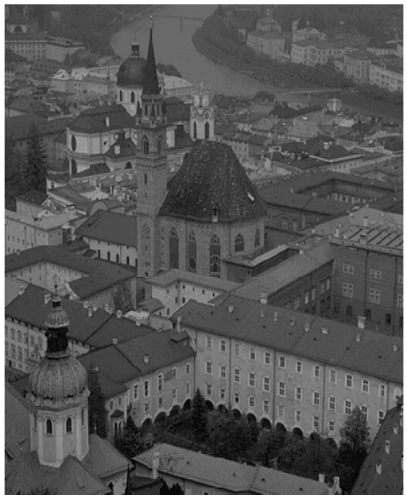
(a) Lena



(b) Baboon

(c) Woman

(d) Bird



(e) Buildings

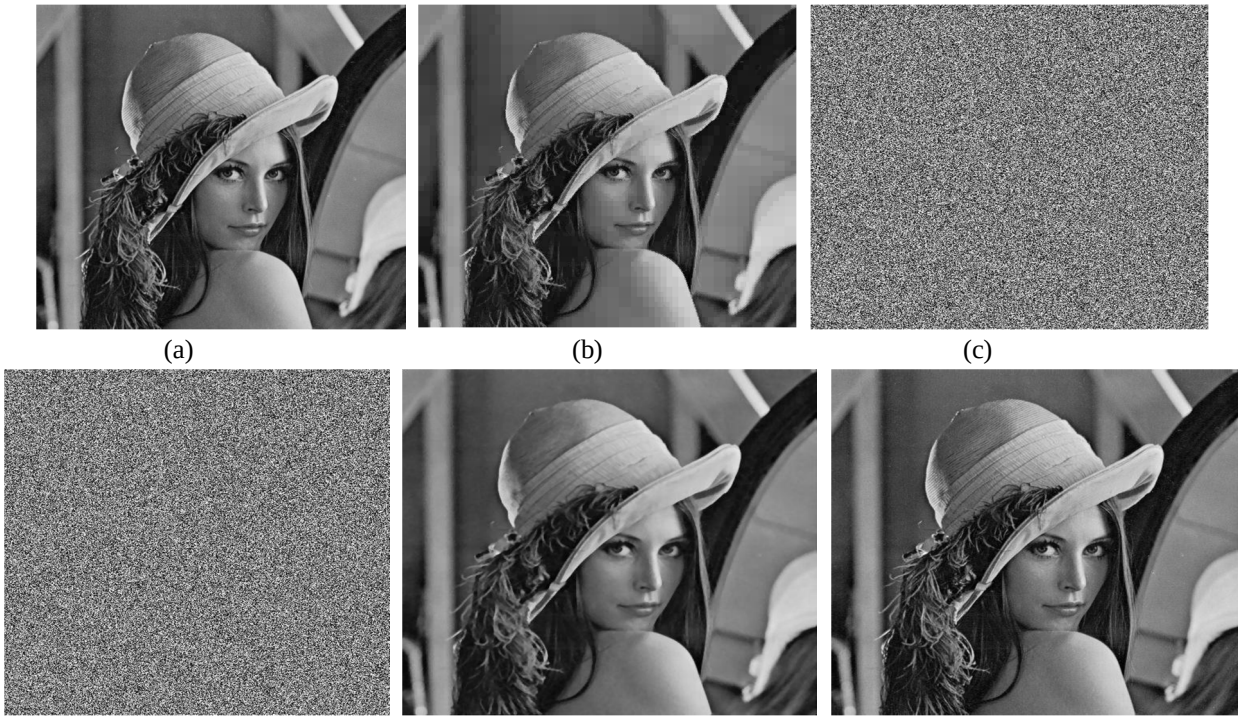
(f) Caravan

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

(g) Food (h) Lake

Fig. 2 512 X 512 JPEG grayscale test images

The experiment is conducted on Lena image as shown in Fig. 3 (a) which consists of a total number of 512 X 512 pixels. First, the image Lena is subjected to quadtree compression which is shown in Fig. 3 (b). In quadtree compression, the input image is compressed to obtain vacant spaces for further data hiding. After the quadtree compression, the image is encrypted using the RC4 encryption algorithm as shown in Fig. 3 (c). The encrypted image is then passed to the data hider. The data hider embeds the secret data into the vacant places using the data hiding key. The marked encrypted image is generated as shown in Fig. 3 (d). Then the marked encrypted image is passed to the receiver. Fig. 3 (e) shows the directly decrypted image and Fig. 3 (f) shows the recovered image.



(a) (b) (c)

(d) (e) (f)

Fig. 3 Results of the proposed method (a) original Lena image, (b) quadtree compressed image, (c) encrypted image, (d) marked encrypted image, (e) directly decrypted image, (f) recovered image

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

The performance of the proposed method is evaluated using PSNR (peak signal to noise ratio) and maximum embedding rate.

$$PSNR = 10log_{10}\frac{255^2}{\frac{1}{MN}\sum_{i=1}^{m}\sum_{j=1}^{N}(O_{i,j} - M_{i,j})^2}$$

The unit of PSNR is decibel (dB). The maximum embedding rate in bpp (bits per pixel) is given by,

$$Embedding\ rate = \frac{Total\ embedded\ bits}{Total\ pixels\ of\ the\ image}$$

Table I shows the maximum embedding rate obtained for each of the test images along with the total number of pixels available for data hiding after conducting the experiment. The total number of pixels denotes the total number of bits that could be embedded.

Table I. Maximum embedding rates obtained for test images

Images	Available pixels	Max. embedding rate (bpp)
Woman	115819	0.4418
Baboon	114979	0.438
Buildings	98512	0.3758
Lena	78857	0.3
Caravan	78259	0.2985
Lake	77955	0.2974
Bird	75391	0.2876
Food	68548	0.2615

The results show that as the number of pixels available for data hiding increases, the embedding rate is more. Here among the test images taken, the standard grayscale JPEG image 'woman' is having larger number of pixels available for data hiding and therefore it has the highest embedding rate comparing to other images. The standard grayscale JPEG image 'food' is having the lowest embedding rate as it has lesser number of pixels available for data hiding comparing to other images.
Table II shows the PSNR results of the directly decrypted image as well as recovered image for the test images taken.

Table II. PSNR Values Obtained for Test Images

Images	Directly decrypted image (dB)	Recovered image (dB)
Woman	35.78	44.24
Baboon	36.12	45.32
Buildings	36.98	46.17
Lena	37.17	47.55
Caravan	37.58	47.89
Lake	37.96	48
Bird	38.12	48.57
Food	39.29	49.83

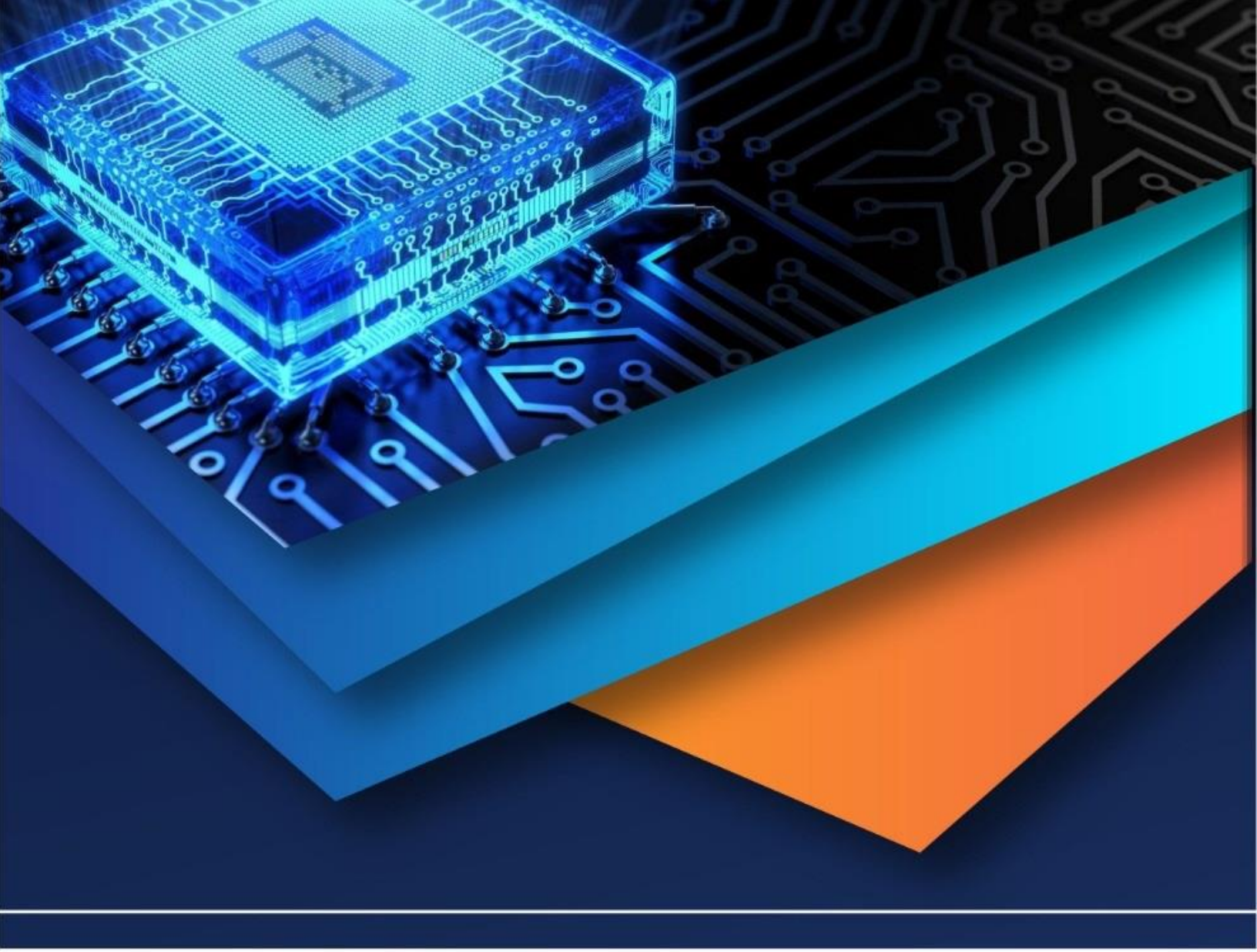
The results show that the PSNR values of recovered images are greater than that of directly decrypted images for all the test images taken. This implies that the recovered image is having higher quality than the directly decrypted image. From table I, the image 'woman' was having higher embedding rate and from table II, the image 'woman' has lesser PSNR values for both directly decrypted image and recovered image comparing to other test images. This is because, as the number of pixels to be modified increases in the recovery step, the prediction error for original pixel restoration is more. The image 'food' was having lower embedding rate but the PSNR values are higher for 'food' image.
The results imply that as the embedding rate increases, the image quality of the recovered image slightly decreases due to the increase in prediction error. But the results also show that the proposed method is successful in achieving a better embedding capacity along with a good range of PSNR values for the recovered images. The computational complexity is calculated using McCabe cyclomatic complexity check and the cyclomatic complexity of the proposed method is found to be 6, which is within the acceptable complexity range.

V. CONCLUSION

A novel approach for reversible data hiding in encrypted JPEG images by using quadtree compression and prediction error has been proposed. The existing techniques vacated space for data hiding after encryption which resulted in increased computational complexity, reduced data hiding capacity and recovered image quality as entropy of the encrypted image is more. To overcome these drawbacks, the proposed method reserved space for data hiding before encryption using quadtree. Then the image is encrypted using RC4 encryption algorithm. Secret data is embedded into the computed space using data hiding key. The data extraction and image recovery are made separable at the receiver side such that the secret data can be extracted without knowing the image contents. Original pixel restoration is done using prediction error technique. The experimental results showed that the proposed method is able to achieve higher embedding rate and improved recovered image quality. Also the simplicity of the quadtree data structure minimised the computational complexity which is proved by McCabe cyclomatic complexity check.

REFERENCES

[1] Tojo Mathew and CI Johnpaul, "Reversible data hiding in encrypted images using interpolation-based distributed space reservation," in Advanced Computing and Communication Systems (ICACCS) 2017 4th International Conference pages 1-6. IEEE, 2017.
[2] Fangjun Huang, Xiaochao Qu, Hyoung Joong Kim and Jiwu Hu, "Reversible data hiding in jpeg images," IEEE Transactions on Circuits and Systems for Video Technology, 26(9):1610-1621, 2017.
[3] Ching-Chun Chang and Chang-Tsun, "Reversible data hiding in jpeg images based on adjustable padding," in 2017 5th International Workshop on Biometrics and Forensics (IWBF), pages 1-6. IEEE,2017.
[4] Zhenxing Qian, Hang Zhou, Xinpeng Zhang and Weiming Zh, "Separable reversible data hiding in encrypted jpeg bitstreams," IEEE Transactions on Dependable and Secure Computing, 15(6): 1055-1067, 2017.
[5] Zhenxing Qian, Haisheng Xu, Xiangyang Luo and Xinpeng Zh, "New framework of reversible data hiding in encrypted jpeg bitstreams," IEEE Transactions on Circuits and Systems for Video Technology, 29(2): 351-362, 2019.
[6] Image manipulation using quadtrees. *https://www.geeksforgeeks.org/image-manipulation-using-quadtrees/*
[7] RC4 encryption algorithm. https://www.geeksforgeeks.org/rc4-encryption-algorithm/
[8] Xiaotian Wu, Wei Sun, "High-capacity reversible data hiding in encrypted images by prediction error," Signal Processing 104 (2014): 387-400.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY