International Journal of Research in Advent Technology, Vol.6, No.8, August 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Enhanced Immense Capacity Reversible Image Hiding with Color Images

N. Thanuja
Department of ECE, JNTU College of Engineering Anantapur-515002
Email: thanujanandi04@gmail.com

Abstract- The Reversible Data Hiding (RDH) is a technique which not only extracts the embedded secret message, but also recovers the original image from the marked image after the message extraction. This concept can also be used for encrypted images. The performance of this technique is evaluated based on the embedding rate and the image quality. Increasing the embedding rate causes more distortion in an image and the quality of an image will be poor. To overcome these drawbacks, This paper introduces a method of an Enhanced Immense Capacity Reversible Image Hiding with Color Images, as an enhancement of the Reversible Data Hiding (RDH). This method provides more data capacity in addition to improved image quality. In addition to that, It introduces the Image Hiding concept in which encrypted image is hided into another image with more security including color images. This method provides less computational complexity and keeps the file structure maximum unchanged.

Index Terms- Reversible data hiding; Image hiding; Encrypted image.

1. INTRODUCTION

As Internet is booming in this generation, sharing confidential information between two parties via internet is always a threat. For that, many security techniques have been developed to protect the information that is transferred through internet. Many Encryption and Decryption algorithms are used for images to enhance security. Here, the confidential information can be transferred secretly through internet by hiding the information in images. Therefore, no hacker including server administrators and others, have access to original message or any other type of transmitted information through public networks.

Data hiding is a process to embed confidential information into an image. In this process, Data invisibility is the major requirement. Whereas Reversible data hiding is a reverse process to the data hiding, this not only extracts the embedded information, but also recovers the original image from the marked image after the information extraction. Reversible data hiding in encrypted images (RDH-EI) is developed based on the reversible data hiding (RDH) concept. Basically, this process includes three persons. They are Content owner, data-hider and Recipient. Content owner is a person who encrypts the original image in to encrypted form. The data-hider embeds secret message in the encrypted image to generate marked encrypted image. Recipient extracts the secret message from the marked encrypted image and recovers the original image by decrypting it [1]. This RDH technique can be developed based on different schemes such as early lossless-compression, integer-transform-based scheme, expansion-based schemes etc., [2]. It is used in many applications like Cloud storage, Covert communication, Image authentication, e-government and secure medical image data system [3].

In Cloud storage, the idea of embedding information into an image helps in saving the storage overhead and also identifying the encrypted image using the information stored in it [4]. In [5], Stream cipher algorithm is used in encrypting the plain text image after that an additional message is embedded into the encrypted image. Here the plaintext image is recovered by decrypting the encrypted image after the extraction of embedded message. To reduce the error rate in the extracted message, this method used the side-match scheme, in which the encrypted image is divided into blocks. Here the message is extracted based on pixel correlation between the blocks [6]. To increase the embedding rate, this method is improved by using the full embedding strategy. In which the image is divided into two parts and the information is embedded in to both the parts [7]. A secret message can also be embedded by modifying the encrypted data in JPEG bit streams [8]. One main drawback in above methods is the extraction of the embedded information. It can be done only after Image Decryption.

To overcome this drawback, Separable RDH-EI was proposed, in which embedded information can be directly extracted from the encrypted image [9]. Here the encrypted pixels are divided into segments and in each segment, LSBs are compressed to embed the additional bits. The LSBs can be recovered only after decryption. Embedding rate can also be increased by using higher bit planes [10]. In order to provide good image quality for higher embedding rates, two

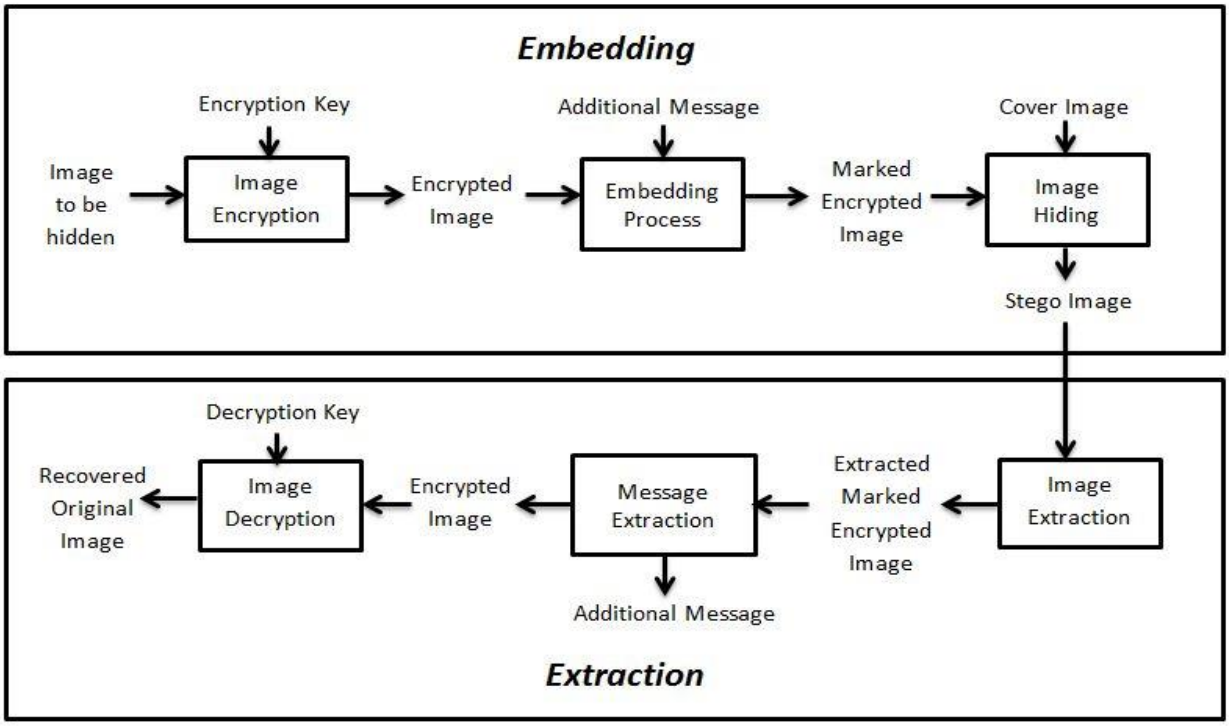

Fig.1. Block diagram of Reversible Image hiding Method

methods are developed based on RDH-EI. They are Joint method and Separable method. In Joint method, both message extraction and image recovery are performed simultaneously. Whereas in Separable method, message extraction and image recovery is performed one after the other.

To evaluate the performance of RDH-EI, embedding rate and distortion are the two important parameters. Increasing the embedding rate causes more distortion in an image. To overcome this problem, a method of an Enhanced immense capacity reversible image hiding with color images is introduced. Till now gray-scale images are used in evaluating different RDH methods. In reality, color images are more popular than the gray ones. So, this method includes color images to increase the embedding performance and improve the image quality.

2. SYSTEM MODEL

The Reversible image hiding method is illustrated in Fig.1. The objective of this method is to develop an application which can hide a secret message in encrypted images and recover the images without any distortion. It includes two phases called Embedding and Extraction.

In Embedding process, first the Image needed to be hidden is encrypted by using encryption key in order to generate an encrypted image. It converts the image into noise like image. After encrypting, this image is divided into three sets for message embedding. In each set, additional message bits are embedded and then the three sets are combined to generate marked encrypted image. Here, More security can be provided by hiding this image in another image using Steganography Concept. This process generates Stego image.

In Extraction process, the marked encrypted image needs to be extracted from the stego image and then additional message bits have to be removed to get the initial encrypted image. At the end, this initial encrypted image is decrypted to recover the original color image without any distortion.

2.1. Image Encryption and Data Embedding

To protect the image contents, Encryption algorithm converts the Original color image I sized $M \times N$ into noise like image. Here stream cipher algorithm is used for encrypting an image. Encrypted image is generated by performing XOR operation of the color image I and encryption key K. Here K is the key stream with $8MN$ bits. It is a set of bits normally required for encryption and decryption. The key stream K is generated by using Pseudo random sequence generator. The sequence which is generated is not completely random, as it is determined by an initial value. The size of the key stream depends on the size of the color image. The encrypted image is generated by,

$$J = Enc(I, K) = I \oplus K \quad (1)$$

To embed the additional message bits, the encrypted image is divided into three sets. Since the

color image contains three channels such as R, G and B channels. In each channel pixels are divided into three sets, they are Square, Circle and Triangle. Such that there are $MN / 4$ pixels in square set, $MN / 4$ pixels in triangle set and $MN / 2$ pixels in circle set as shown in Fig.2.

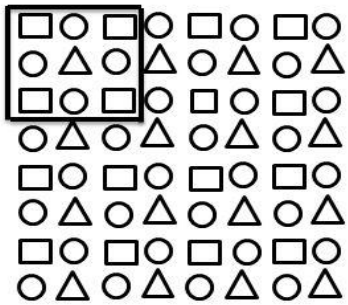

Fig.2. Three sets of an encrypted image

Here the pixels are divided by taking a 3×3 matrix of an image and considering corner pixels as square set, edge pixels as circle set and center pixel as triangle set. This process should be repeated for each channel in an image. After separating the three sets, additional message bits need to be embedded in to any one set of each channel. Additional message bits are generated randomly for each set and embedding is performed in each channel by using Histogram pair concept.

Histogram of an image $h(x)$ is defined as the probability of occurrence of pixels with one certain value. In Histogram pair method, Consider two integers a and b where $x \in [a, b]$. Let $h(a) = m$ and $h(b) = 0$, these points are considered as a histogram pair and it is denoted by $h = [m, 0]$ as shown in Fig.3.

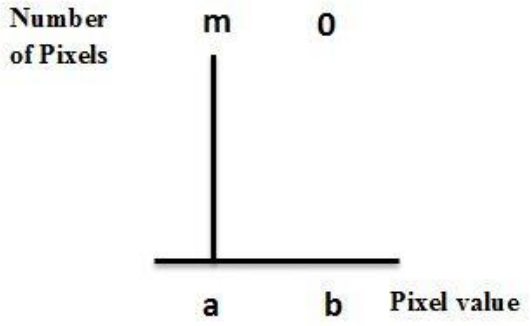

Fig.3. Peak point and zero point of a histogram

In the histogram, find the zero point whose pixel value with 0 number of pixels and then find the peak point, whose pixel value with maximum number of pixels. After finding these points, increment the pixel values by 1 unit within the range $[a, b]$. After incrementing, check the additional message bits to be embedded. If the message bit is 0, decrement the pixel value within the frequency range by -1 unit. If the message bit is 1, then there is no need to increment the pixel value. For example, Let us consider an image with pixel values $X = [p\ p\ p\ p]$ and the message to be embedded is $D = [1\ 0\ 1\ 0]$. In message embedding, if the message bit is 1, the pixel value is incremented by 1. The change in pixel values is represented by q. If the message bit is 0, there will be no change in pixel value. After the message embedding, the pixels values of the image changes from $[p\ p\ p\ p]$ to $[q\ p\ q\ p]$. Repeat this process for each and every bit. These message bits have to be embedded in any one set of each channel. After the completion of embedding, these R, G and B channels have to be combined to generate the Marked Encrypted Image.

An image sent in secret (steganography) in an encrypted form is more secured than a clear encrypted image. In order to provide the security, the marked encrypted image is hided in another image. Here for hiding an image, LSB steganography method is used. For this purpose, first select any image and consider it as a cover image. To hide the marked encrypted image, increase the size of a cover image as required. After that, clear the LSB bits of the cover image and move the MSB bits of the marked encrypted image into those LSB bits. And then clear the LSB bits of marked encrypted image. Insert the marked Encrypted image into the cover image by dividing it into four quadrants. By inserting the marked encrypted image in to cover image, Stego image is generated. Embedding process will be completed after generating the stego image.

2.2. Image Extraction and Recovery

Extraction is exactly the reverse process of Embedding. In the Extraction, First remove the additional message bits embedded in the Encrypted image in order to recover the original color image. For this purpose, Marked encrypted image is extracted from the stego image. Clear the LSB bits and resize the stego image as required so that it becomes original image's resolution. After that extract the marked encrypted image by shifting the MSB bits in the quadrants. Now, divide the marked encrypted image in to the Square, the Circle and the Triangle sets again.

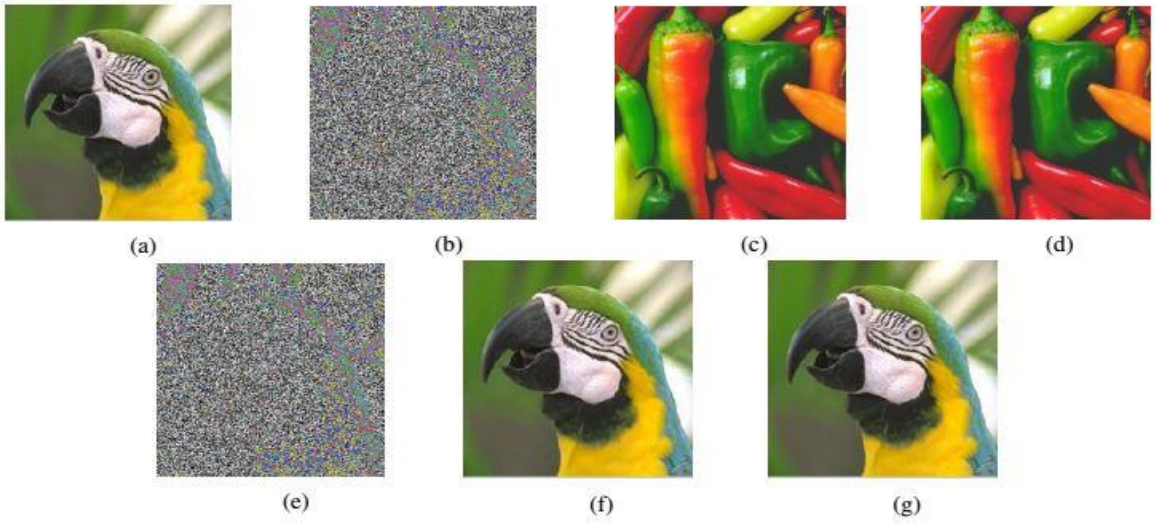

Fig.4. Experimental results of Reversible image hiding, (a) original image (parrot), (b) marked encrypted image, (c) cover image (chillies), (d) stego image,(e) extracted marked encrypted image (f) directly decrypted image, and (g) losslessly recovered image.

Using the Histogram pair method, find the zero and peak points once again. After finding the points, compare the pixel values of the marked encrypted image with the pixel values of the Original image. If the value is incremented in the marked encrypted image then retrieve the message bit as 1 and if there is no increment in the value of the marked encrypted image then retrieve the message bit as 0. For example, Let us consider the original image with pixel values $X = [p\ p\ p\ p]$ and the marked image with pixel values $Y = [q\ p\ q\ p]$. In message extraction, compare the pixel values of both the images. If the pixel value is changed from p to q, then retrieve the message bit as 1. If there is no change in pixel values, then retrieve the message bit as 0. After the message extraction, the extracted message is $D = [1\ 0\ 1\ 0]$. Repeat this process for all pixel values. After that shift the histogram by -1 unit within the range $[a, b]$. Repeat this process for each and every channel and then combine the R, G and B channels to get the initial encrypted image.

Now, Original color image can be recovered by performing XOR operation of recovered encrypted image J and encryption key K. Here K is the key stream with $8MN$ bits. This is a set of bits normally used in cryptographic algorithms. Here the key stream K is generated by using Pseudo random sequence generator. The key which is used for encryption can also be used for decryption. The original color image is recovered by,

$$\mathrm{I} = Dec(J, K) = J \oplus K \qquad (2)$$

3. EXPERIMENTAL RESULTS

A group of experimental results are shown in Fig. 4, in which Fig. 4(a) is the color image (parrot) sized $255 \times 255 \times 3$. This image is encrypted by using encryption key to generate encrypted image. This image is used to embed three additional messages in R, G and B channels. After embedding, Fig. 4(b) Marked encrypted image is generated. To increase the security of an image, this marked encrypted image is hided in a cover image (chillies) Fig. 4(c). By hiding the image, Fig. 4(d) stego image is generated. To recover the original image, Fig. 4(e) marked encrypted image is extracted from the stego image. Fig. 4(f) shows the image generated by direct decryption Fig. 4(e). From the marked encrypted image, embedded message bits can be extracted without any error. The original color image recovered by the decryption is shown in Fig. 4(g) which is same as Fig. 4(a).

This method also introduces the novelty in evaluation parameters from the previous methods by adding more performance parameters such as PSNR, MSE, SSIM, Embedding Rate and Error Rate. Here the experimental results are carried out by using various color images and the performance parameters are calculated for every image. Usually the Quality of an image depends on the Embedding rate. So to evaluate the Image quality, these performance parameters have to be calculated for increasing Embedding rates. Since we are experimenting this method on Color images, Instead of embedding one additional message, three additional messages can also be embedded in each channel of an image. The Embedding rate for adding three additional messages is more when compared to the embedding rate for adding one additional message. In Fig.4., original color image is used to embed 64948 bits (0.33 bpp) additional message into the encrypted image. The PSNR of the recovered image Fig.4(g) is equal to 43.08 dB. Similarly, the PSNR of the same image for

Table 1. Performance metrics of different images for embedding one message

Images to be hidden	PSNR (dB)	MSE	SSIM	Error rate
Parrot	43.0847	3.1961	0.9964	0.0019
Lena	42.1055	4.0043	0.9957	0.0020
Barbara	45.1265	1.9973	0.9982	0.0019

embedding three additional messages with an embedding rate (0.99 bpp) is 37.10 dB. This recovered image preserves good quality even for high embedding rate. In addition to this, SSIM is also calculated for better Image quality.

Table 2. Performance metrics of different images for embedding three messages

Images to be hidden	PSNR (dB)	MSE	SSIM	Error rate
Parrot	37.1022	12.6725	0.9863	0.0088
Lena	36.3184	15.1788	0.9832	0.0084
Barbara	39.0237	8.1416	0.9933	0.0079

Table 1 shows the PSNR, MSE, SSIM and Error Rate of different Color images for adding one additional message. Here Errors may occur during the Image recovery. These errors can be analyzed by calculating the Error rate. It is the ratio of incorrectly recovered bits to the total number of bits in an Image. These incorrectly recovered bits are identified by comparing the bits of Original color image with the bits of recovered image. Similarly, Table 2 shows the PSNR, MSE , SSIM and Error Rate of different Color images for adding three additional messages. Since the embedding rate is high, the values are slightly decreased when compared with Table 1.

Table 3. Comparison of PSNR for different Embedding Rates

No. of embedded bits	Embedding rate (bpp)	PSNR (dB)		
		Parrot	Lena	Barbara
11350	0.17	36.84	36.28	38.74
25760	0.39	36.73	36.79	38.84
43570	0.67	39.24	36.34	39.23
64948	0.99	36.86	36.79	38.74

In order to achieve high embedding rates, different additional messages with increasing number of bits are embedded in the different images. Here Table 3 shows the comparison of PSNR for increasing Embedding rates. For this purpose, Additional messages with increasing number of bits are embedded in the different color images. Fig.5. shows the PSNR of different images corresponding to increasing Embedding rates. Even for high embedding rate, this method maintains better Image quality.

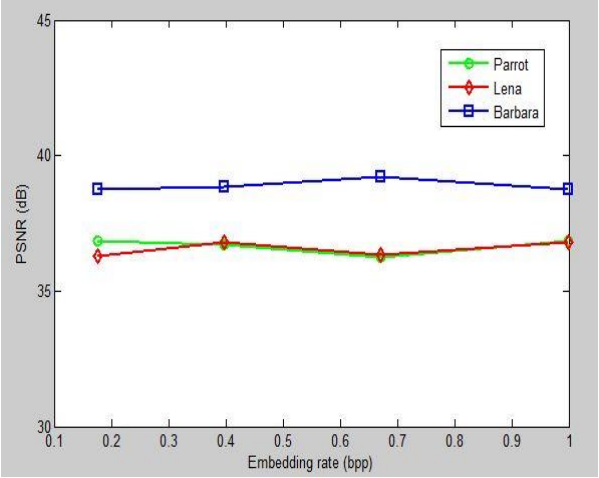

Fig.5. Embedding Rate – Distortion Comparisons

4. CONCLUSION

In this paper, a novel Reversible image hiding method for colored images with low computational complexity is introduced, which contains Image Encryption, Message Embedding and Image Recovery steps. Previous methods have implemented several techniques to embed the data in Gray scale images only. Here the Original Color image is entirely encrypted by using stream cipher algorithm and then additional message bits are embedded into an encrypted image using Histogram pair method. To provide more security, Steganography concept is used in which the encrypted image is hided in another image. The original color image is recovered by removing additional message bits and decrypting the encrypted image using the decryption key. Finally, Original color image was recovered and the Quality of the recovered image is analyzed by evaluating parameters such as PSNR, MSE, SSIM and Error rate. This method helps in increasing the data capacity in addition to the improved Image quality.

References

[1] Z. Qian, X. Zhang and G. Feng, "Reversible Data Hiding in Encrypted images based on Progressive recovery", *IEEE Signal processing Letters*, Vol.23: 1672-1676, 2016.

[2] X. Li, W. Zhang, B. Ou and B. Yang, "A brief review on reversible data hiding: current techniques and future prospects", *IEEE China summit and International Conference on Signal and Information Processing*, 426-430, 2014.

[3] H. Wang, W. Zhang and N. Yu, "Protecting Patient Confidential Information based on ECG Reversible data hiding", *Multimedia Tools and Applications*, doi: 10.1007/s11042-015-2706-2, 2015.

[4] Z. Fu, X. Sun, Q. Liu, et al. "Achieving efficient cloud search services: Multi-Keyword ranked search over encrypted cloud data supporting parallel computing", *IEICE Transactions on Communications*, 98(1): 190-200, 2015.

[5] X. Zhang, "Reversible data hiding in encrypted images", *IEEE Signal processing Letters*, 18(4): 255-258, 2011.

[6] W. Hong, T. Chen and H. Wu, "An improved reversible data hiding in encrypted images using side match", *IEEE Signal Processing Letters*, 19(4): 199-202, 2012.

[7] M. Li, D. Xiao, A. Kulsoom, and Y. Zhang, "Improved reversible data hiding for encrypted images using full embedding strategy", *Electronic Letters*, 51(9): 690-691, 2015.

[8] Z. Qian, X. Zhang and S. Wang, "Reversible data hiding in encrypted JPEG bitstream", *IEEE Transactions on Multimedia*, 16(5): 1486-1491, 2014.

[9] X. Zhang, "Separable reversible data hiding in encrypted image", *IEEE Transactions Information Forensics and Security*, 7(2): 826-832, 2012.

[10] X. Wu and W. Sun, "High capacity reversible data hiding in encrypted images by prediction error", *Signal Processing*, 104: 387-400, 2014.