International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Watermarked and Encrypted Image for Reversible Data Hiding by Reserving Room Algorithm with Random Pixel Data Hiding

Paul Jose[1], Anil C. B[2]

[1]Sree Narayana Gurukulam College of Engineering, Kadayirippu, Ernakulam, Kerala, India

[2]Sree Narayana Gurukulam College of Engineering, Kadayirippu, Ernakulam, Kerala, India

Abstract: ***Reversible Data Hiding (RDH) gaining the attraction in the world of secured message passing through encrypted images. It is due to the excellent property of recovering the original cover or image after embedded data is extracted while protecting the encrypted image content's confidentiality. The related works may embed data by reversibly vacating room from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. In this paper, we propose a novel method by reserving room before encryption and a watermarking technology with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image with an authentication to the sender. The data which is more secure and unrecoverable for an external hacker/intruder since it is stored in a random way. The proposed method can get 100% reversibility with data extraction and image recovery. Experiments show that this novel method can embed more than 10 times as large payloads for the same image quality as the earlier methods, such as for PSNR 40dB or above.***

Keywords: Reversible data hiding, image encryption, random data hiding, self-embedding.

1. Introduction

Reversible data hiding in images is a technique, by which the original image can be recovered without any loss in quality after the data embedded in an encrypted image is extracted. This technique is widely used in medical imagery, military imagery and forensics, where no distortion of the original cover image is allowed to avoid data loss and for the smooth message passing. Reversible data hiding was introduced in the beginning of 2000's and gained the research interest and have emerged in recent years with some general frameworks.

In theoretical aspect a rate-distortion model for Reversible data hiding [1], proved the rate-distortion bounds for less memory and proposed a recursive code construction, however, does not approach the bound. An improved recursive code construction [2],[3] for binary covers, which establishes the equivalence between data compression and Reversible data hiding for binary covers. A more popular method, difference expansion[5], where the difference of each pixel group is expanded thus the least significant bits of the difference are all-zero and can be used for embedding messages. Another strategy is histogram shift [6], where space is saved for data embedding by shifting the bins of the histogram of gray values. The combined difference expansion or histogram shift [7]-[11] to residuals of the image, to achieve better performance. Encryption [12] is an effective and popular means as it converts the original and meaningful content to incomprehensible one. Suppose a medical image database is stored in a data center, and a server in the data center can embed notations into an encrypted version of a medical image through a RDH technique. With the notations, the server can manage the image or verify its integrity without having the knowledge of the original content, and thus the patient's privacy is protected. On the other hand, a doctor, having the cryptographic key, can decrypt and restore the image in a reversible manner for the purpose of further diagnosing. Some attempts on RDH in encrypted images have been made such as divided the encrypted image into several blocks. The data extraction and image recovery proceed by finding which part has been flipped in one block. But the encrypted image [18] should be decrypted first before data extraction. To separate the data extraction from image decryption, emptied out space for data embedding following the idea of compressing encrypted images. Compression of encrypted data can be formulated as source coding with side information at the decoder, in which the typical method is to generate the compressed data in a lossless manner by exploiting the syndromes of parity-check matrix of channel codes. In Reversible Data Hiding, only the sender and receiver are aware of the hidden data and typically if the loaded file falls into the hands of anyone else they wouldn't retrieve the hidden data. While seeing an encrypted Image the first thing that comes to a third person is that, an Image encrypted and this image is the message that is exchanged between the sender and receiver so decryption of Image is done. Here in this thesis the encrypted image is watermarked to provide authentication and even they realized there is a message in the Image it is not possible to recover it so that each character is represented as 8 bits and that too is randomly hidden in the image with a seed provided.

2. Related Work

The methods proposed in [16]–[18] can be summarized as the framework, "vacating room after encryption (VRAE)", as illustrated in Figure 1(a).In this framework, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider and the data hider can embed some auxiliary data into the encrypted image by losslessly vacating some room according to a data hiding key.

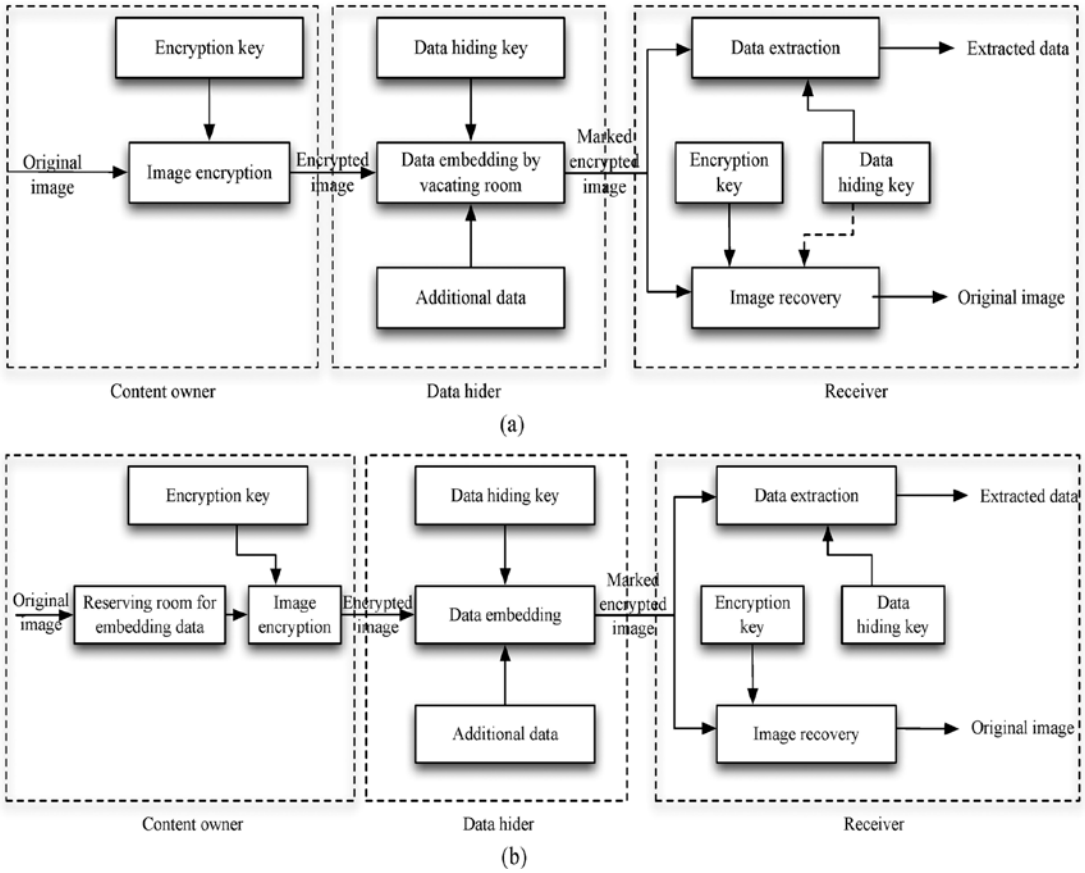

Figure 1: Previous works, (a) showing without reserving room (b) showing with reserving room before encryption

Then a receiver, maybe the content owner himself or unauthorized third party can extract the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key. In all methods of [16]–[18], the encrypted 8-bit gray-scale images are generated by encrypting every bit-planes with a stream cipher. The method in [16] segments the encrypted image into a number of non-overlapping blocks sized by a x a each block is used to carry one additional bit. Additional bit is considered as 0.To do this, pixels in each lock are pseudo-randomly divided into two sets S1 and S2 according to a data hiding key. To have decode or to recover the image they gets into trouble while doing mode function when reached 8 the value coincides. Another risk of defeat of bit extraction and image recovery when divided block is relatively small (e.g., a=8) or has much fine detailed textures.

As shown in the figure 1(b) its holds good enough to have good recovery capability and data exactly recovered [19]. Reduced the loss of bit to the minimum, by reserving the pixels before the image is encrypted. It also allows the data to be extracted before and after the image is being decrypted. They too failed to have the security for the data what is hidden and also to have an authentication between the sender and the receiver. In the proposed method since the data hider is getting a chance of using the encrypted image, an authentication is produced in terms of watermarking. After the data is hidden there is a chance for attackers to get the sequentially hidden data. The proposed method explains how to hide data or message in random manner and also to store a single character in eight different parts or pixels using ASCII to 8-bit data conversion.

3. Proposed Method

In the proposed method the pixels in the image is vacated according to the number of data to be hidden, which is defined as reserving room. The image is then encrypted using some algorithms. An authentication is added to the image uses watermarking technology, the watermarked image is then send to the data hider to hide the data. Using a random key a randomization of data is done for each character which is represented in 8 bits, they are then randomly hidden in the vacated space in the image using reserving room algorithm and send to the receiver. The receiver first checks for the authentication which means recovering the watermark. Then using the random key the data is retrieved. Or in other case the image is decrypted and the data is retrieved**.**

3.1Reserving Room for Embedding Data

Here the steps for reserving pixel (room) for embedding data

1. Enter the maximum number of characters going to hidden (m) [20].
2. Using the value of m Image partition is done.
3. Before going for partition the most complex part of the image is found using the first order smoothness function (f)

$$f = \sum_{u=2}^{m} \sum_{v=2}^{N-1} |Cu, v - \frac{Cu-1, v + Cu, v + Cu, v-1 + Cu, v+1}{4}$$

4. Higher f relates to blocks which contain relatively more complex textures, f+m is selected as A part of the image and rest as part B.

5. The Least Significant bits (LSB) of B is calculated and the least significant bits (LSB) of A replaces it.
6. The first 48 bits of A is stored for the value m, the histogram shifting values, the pixels there the data to be hidden.
7. Thus Self Reversible embedding is done by A on the top and then B.

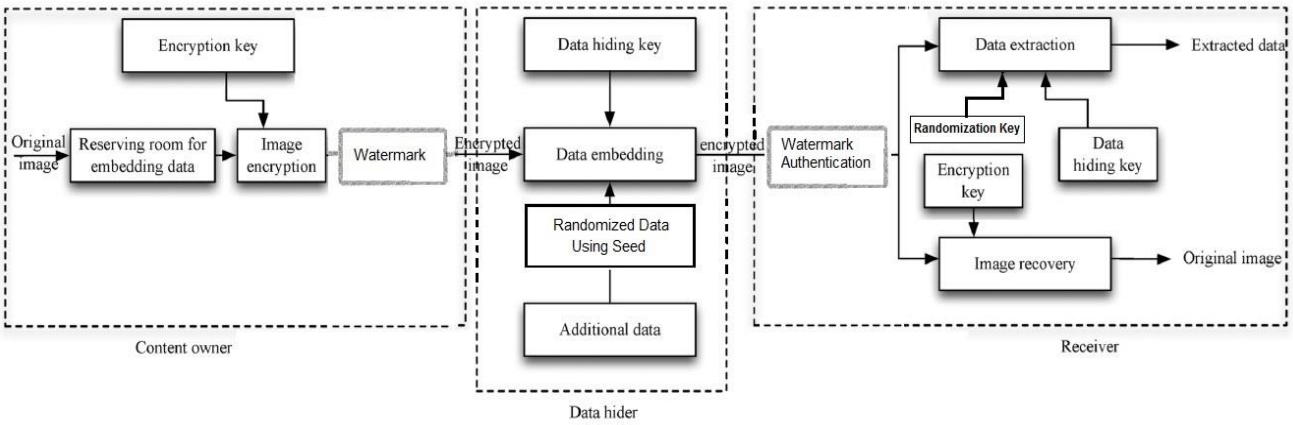

Figure 2: Proposed Method

3.2 Image Encryption

1. A gray value X*i,j* ranging from 0 to 255 can be represented by 8bits, X*i,j*(0), X*i,j*(1),….. X*i,j*(7) such that[20]

$$Xi,j(k)=\frac{X_{i,j}}{2^k} \bmod 2,\ k=0,1,\ldots.7$$

2. The encrypted bits E*i,j*(k) can be calculated through exclusive-or operation

$$Ei,j(k)=Xi,j(k)\oplus ri.j(k)$$

3.3. Data Embedding

1) Data is embedded in the A part of the self-embedded image [20].
2) Data as characters are converted to ASCII
3) This single ASCII value which has got an integer value is then converted to 8 bit binary values 0 and 1.

3.4. Randomization of data for hiding

1) A single character is represented in 8 bits and are randomly arranged.
2) A word which is used for randomization can be separated as even characters and odd characters.
3) All the even Characters are converted to ASCII and added together.
4) All the odd characters are converted to ASCII and subtracted together.
5) The absolute value of each is find and the greater value is submitted to the random function to randomize the data.

3.5. Authentication Signature using Watermarking.

1) Apply moment-preserving thresholding [19].
2) Obtain two representative gray values g1 and g2
3) 3. The value above gavg is considered as 1 and the value below gavg is considered as 0.

3.6. Extracting Data from Encrypted Images

1) An inferior database manager may only get access to the data hiding key
2) Manipulate data in encrypted domain.
3) The database manager gets the data hiding key
4) He can decrypt the LSB-planes of the A part and extract the additional data by directly reading the decrypted version.

3.7. Extracting Data from Decrypted Images

1) In this case both the image and the data is recovered.
2) The Encrypted Image is decrypted using the encryption key.
3) Thus the image gets encrypted.
4) The data is stored from the 49th bit of the A part and can be recovered from it by using the information passed about the LSB plane.

4. Experimental Analysis and Results

Figure 3: Input image

Figure 4: Self-embedded

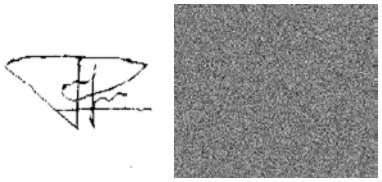

Figure 4: Watermark **Figure 5**: Watermarked and Encrypted Image

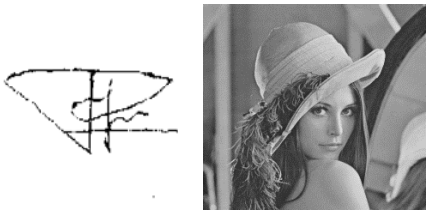

Figure 6: Recovered **Figure7**: Recovered Image Watermark

The PSNR value for the resolution of 512X512 image which varies in number of characters hidden. As the number of characters increased the time for extraction and also the PSNR value gets increased

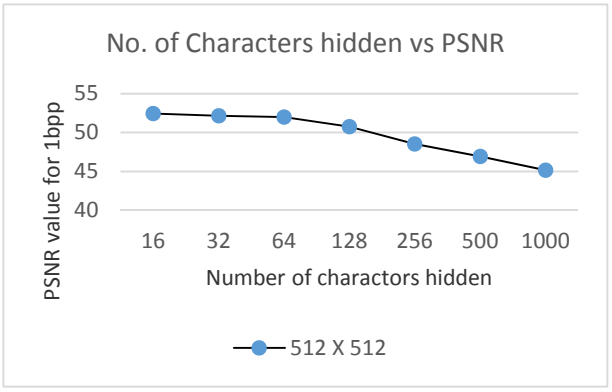

Figure 8: No. of Characters hidden vs PSNR

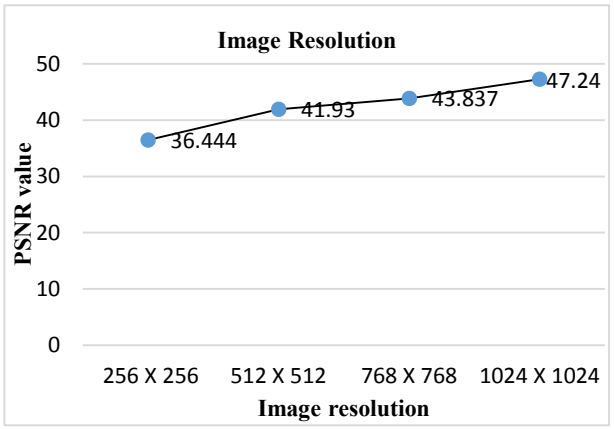

Figure 9: Variation of different Resolutions Image vs PSNR (With number of Characters = 2000)

The above analysis is done for an image with fixed resolution to get the results of the variation in PSNR value. From the above table and the figure it is analysed that

- As the number of characters hidden increased the PSNR value gets decreased.
- As the image resolution increases the PSNR value gets increased.
- Here the analysis is done by the same image with doing changes in the resolution, with same encryption key and randomization key.
- For a fixed image with same resolution with differ in number of characters, PSNR value get decreased by increase in number of characters.
- More data in low resolution image gives small PSNR value.

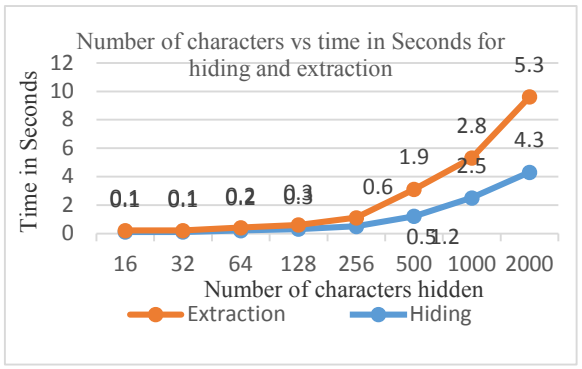

Figure 10: Number of characters vs time in seconds

a) Time for the data hiding and data extraction increases by different factor
   - Number of Characters.
   - Randomization methodology.
   - Length of randomization key.
   - Size of the Image.
b) The above shown results are of in same resolution (512 X 512), same randomization key and same encryption key.
c) It is clear from the analysis that as the number of characters increased the time for data hiding and data extraction gets increased.
d) The time is not a matter for the sender; it is only affected by the data hider during the data hiding, and for receiver for extracting the message and also extracting the image.
e) By this result it is clear for the security system that it takes only less than 5 seconds to send a message of character length 2000.
f) There is only a small difference in the time for hiding the data and extraction of the data with a fixed character length.

5. Conclusion

Previous methods implement RDH in encrypted images by vacating room after encryption; have got less security and authentication in reserving room before encryption. Thus the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images.

The Randomized data hiding made it more secure. Even though the data that is needed is to be secret they stored sequentially will be able for an intruder or a hacker to get the data with similarities on the bit pattern. Since here it is a random method eight bits of each character is stored in different places of the image. Time gets a real attention in this project that can yield good result which defines only less than 5 seconds for hiding a message or a paragraph of message of 2000 characters.

Watermarking gives a special authentication between sender and receiver. Though a chance of data hider to edit or

change the image can be perfectly removed here. Both the implemented methods are found to be good yielding good results and can be used for the real time applications.

6. Future Work

Randomization can be done more accurately. Here is negligible bit loss in the B part of the self-embedded image which can be avoided by redundancy avoiding pixels. Time for data hiding and extraction can be still reduced.

References

[1] T. Kalker and F. M. Willems, "Capacity bounds and code constructions for reversible data-hiding", *in Proc. 14th Int. Conf. Digital Signal Processing (DSP2002),* 2002, pp. 71–76.
[2] W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes for reversible data hiding", *in Proc 13th Information Hiding (IH'2011)*,LNCS 6958, 2011, pp. 255–269, Springer-Verlag.
[3] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers", *IEEE Trans. Image Process.,* vol. 21, no. 6, pp. 2991–3003, Jun. 2012.
[4] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats", *in Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.
[5] J. Tian, "Reversible data embedding using a difference expansion", *IEEE Trans. Circuits Syst. Video Technol*., vol. 13, no. 8, pp. 890–896, Aug. 2003.
[6] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
[7] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking", *IEEE Trans. Image Process*.,vol.16,no.3, pp. 721–730, Mar. 2007.
[8] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection*", IEEE Trans. Image Process*., vol. 20, no. 12, pp. 3524–3533, Dec. 2011.
[9] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting" , Signal Process., vol. 89, pp. 1129–1143, 2009.
[10] L. Luo et al., "Reversible image watermarking using interpolation technique", *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.
[11] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversible watermarking algorithm using sorting and prediction" , *IEEE Trans. Circuits Syst. Video Techn*ol., vol. 19, no. 7, pp. 989–999, Jul. 2009.
[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC, 1996.
[13] K. Hwang and D. Li, "Trusted cloud computing with secure resources and data coloring*", IEEE Internet Comput*., vol. 14, no. 5, pp. 14–22, Sep./Oct. 2010.
[14] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data", *IEEE Trans*. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.
[15] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images", *IEEE Trans. Image Process.*, vol.19,no.4, pp. 1097–1102, Apr. 2010.
[16] X. Zhang, "Reversible data hiding in encrypted images,", *IEEE Signal Process. Lett*., vol. 18, no. 4, pp. 255–258, Apr. 2011.
[17] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match*,", IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.
[18] X. Zhang, "Separable reversible data hiding in encrypted image", *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.
[19] W. H. Tsai, "Moment-preserving thresholding: A new approach", *Comput. Vis. Graph. Image Process.*, vol. 29, no. 3, pp. 377–393, Mar.1985.
[20] Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu, and Fenghua Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption", *IEEE Transactions On Information Forensics And Security*, Vol. 8, No. 3, March 2013

Author Profile

Paul Jose received the B.E. Degree in Computer Science and Engineering from Angel College of Engineering and Technology, Anna University, Tamil Nadu, in 2012 and pursuing M.Tech. Degree in Computer Science and Engineering from Sree Narayana Gurukulam College of Engineering, M.G. University, Kerala, during 2012-2014.

Anil C B received the B.Tech Degree in Computer Science and Engineering from M.A. College of Engineering, M.G. University, Kerala, in 2002 and M.Tech degree in Software Engineering from Cochin University of Science and Technology in 2007. Currently pursuing Ph.D. in Sensor networks at Cochin University of Science and Technology. Now working as Assistant Professor in Sree Narayana Gurukulam College of Engineering, Kerala, India