

## Sharing Memory before Encryption for Defended Reversible Data Hiding In Encrypted Images

Sonali Jambhulkar<sup>1</sup>, Mona Mulchandani<sup>2</sup>, Priyanka Dudhe<sup>3</sup>

Dept. of Computer Science & Engg., Jhulelal Institute of Technology, Nagpur-INDIA  
Email: [sonali.jambhulkar7@gmail.com](mailto:sonali.jambhulkar7@gmail.com), [mona.mulchandani@jit.org.in](mailto:mona.mulchandani@jit.org.in), [p.dudhe@jit.org.in](mailto:p.dudhe@jit.org.in)

**Abstract:** RDH is a high-protection technology used to hide secret information inside the image and can recover the original image and secret information completely. Bitmap image and injected information scheme have a number of collective media applications that are needed. Now a days, the more attraction is towards reversible data hiding (RDH) in encoded images, as it continues to have the highest quality property that the original cover can be recovered without distortion after the injected information has been removed while protecting the privacy of the image content. All earlier methods inject information by emptying room from the encrypted images reversibly, which may result in distortion of information withdrawal or image restoration. In this paper, we planned another strategy in which, using our new technique, we simply encode an image without its header. It is simple for the information hider to reversibly inject information into the encoded image by using this new technique. The proposed technique can obtain real reversibility, i.e. withdrawal of information and restoration of images is free of distortion.

**Keywords:** Encryption, Decryption and Reversible Data hiding

### I. Introduction

The role of privacy and security is very important. Protecting our secret data is very important. Internet communication is becoming more frequent on a day-to-day basis. Information security has become a critical issue because of a large number of threats to communications security. Systems for data embedding and data hiding are needed to address some of the major challenges posed by the widespread distribution of multimedia content across digital communication networks. Hiding data is a process in which data is hidden in cover media. It connects two data sets such as embedded data set and another set of media cover data. Different applications characterize the relationship between these two data sets. In particular, these systems enable technologies for (1) enforcement and protection of copyright (2) authentication and detection of multimedia signal and image manipulation. In theoretical terms, Kalker and Willems[1] established a rate-distortion model for RDH by demonstrating RDH's rate-distortion bound for memory-less covers and proposing a recursive code construction that, however, does not approach the bound. Zhang et al.[2],[3] improved the recursive code construction for binary covers and demonstrated that this construction could achieve the rate distortion bound as long as the compression algorithm reaches entropy, which establishes the equivalence for binary covers between information compression and RDH.

### II. Related Work

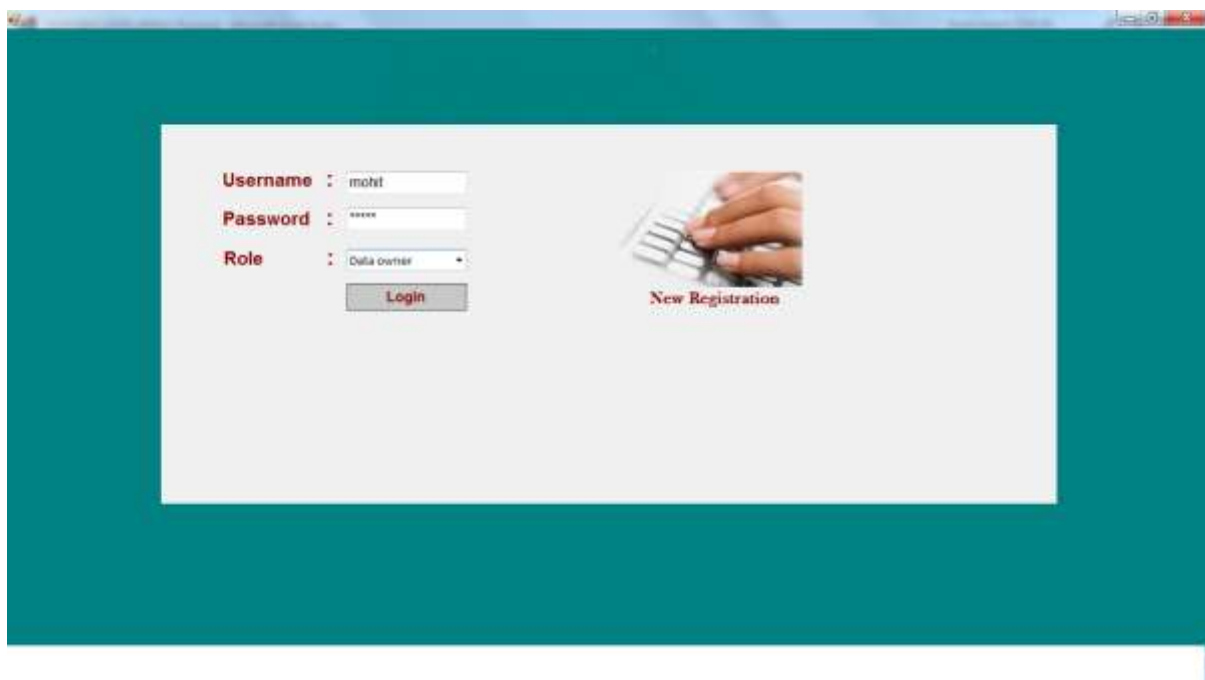
In practical terms, over the past few years, many RDH techniques have emerged. A general framework for RDH was constructed by Fridrich et al.[4]. Spare space can be saved to embed extra information by first extracting compressible features of the original cover and then losslessly compressing them. Another RDH strategy is the histogram shift (HS)[5] that saved space for data embedding by shifting the bins of histogram of gray values. Zhang split the encrypted image into multiple blocks in[6]. Space can be vacated for the embedded bit by flipping 3 LSBs of half the pixels in each block. Data extraction and restore image proceed by finding which part in one block was flipped. This process can be realized in decrypted image using spatial correlation. Hong et al.[7] improved the decoder-side method of Zhang by using a different estimation equation and side match technique to further exploit the spatial correlation to achieve much lower error rate. These two above-mentioned methods depend on the original image's spatial correlation to gain data. That is, before extracting information, the encrypted image should be decrypted first. To separate data extraction from image decryption, Zhang[8] has emptied data embedding space after compressing encrypted images[9],[10]. Compression of encrypted data can be formulated as source code with side information in the decoder[9], whereby the typical method is to generate the compressed data in a lossless manner by exploiting parity check matrix of channels code. Our goal is to develop a secure system for sending data over a network consisting of separate and reverse

image encryption (such as bmp image and txt file encryption), data embedding that prevents access to secret data by any third party. In this method, we can achieve real reversibility with the help of symmetric key, that is, data extraction and image recovery are free of distortion.

### **III. Current Implementation**

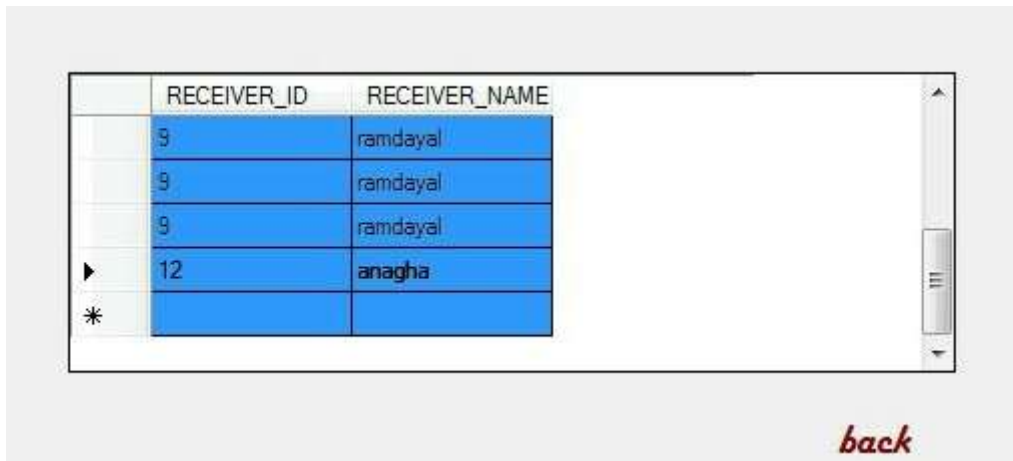
The proposed work is made up of image encryption, data embedding and data-extraction/image-recovery phases. This projected technique can accomplish real reversibility, that is, data extraction and image recovery are free of any error. This method proposes a separable reversible data hiding in encrypted image. In the proposed method, the original image is encrypted using an encryption key and the additional data are embedded into the encrypted image using data-hiding key. With an encrypted image containing additional data, if the receiver has only the data-hiding key, receiver can extract the additional data though receiver does not know the image content. If receiver has only the encryption key, receiver can decrypt the received data to obtain an image similar to the original one, but cannot extract the embedded additional data.

*Sharing Memory before Encryption For Defended Reversible Data Hiding in Encrypted Images*



The first page is login page. whenever the user wants to login as a data owner or receiver then he can directly enter his username and password if he has already registered otherwise he need to click on new registration and fill all the details and click on submit button. Once the submit button is clicked, data will be saved in database.

Below is the screenshot of the data saved in the database.



A screenshot of a web application interface showing a table with two columns: RECEIVER\_ID and RECEIVER\_NAME. The table contains five rows of data. The first three rows have RECEIVER\_ID 9 and RECEIVER\_NAME ramdayal. The fourth row has RECEIVER\_ID 12 and RECEIVER\_NAME anagha. The fifth row is empty and marked with an asterisk. A 'back' button is visible below the table.

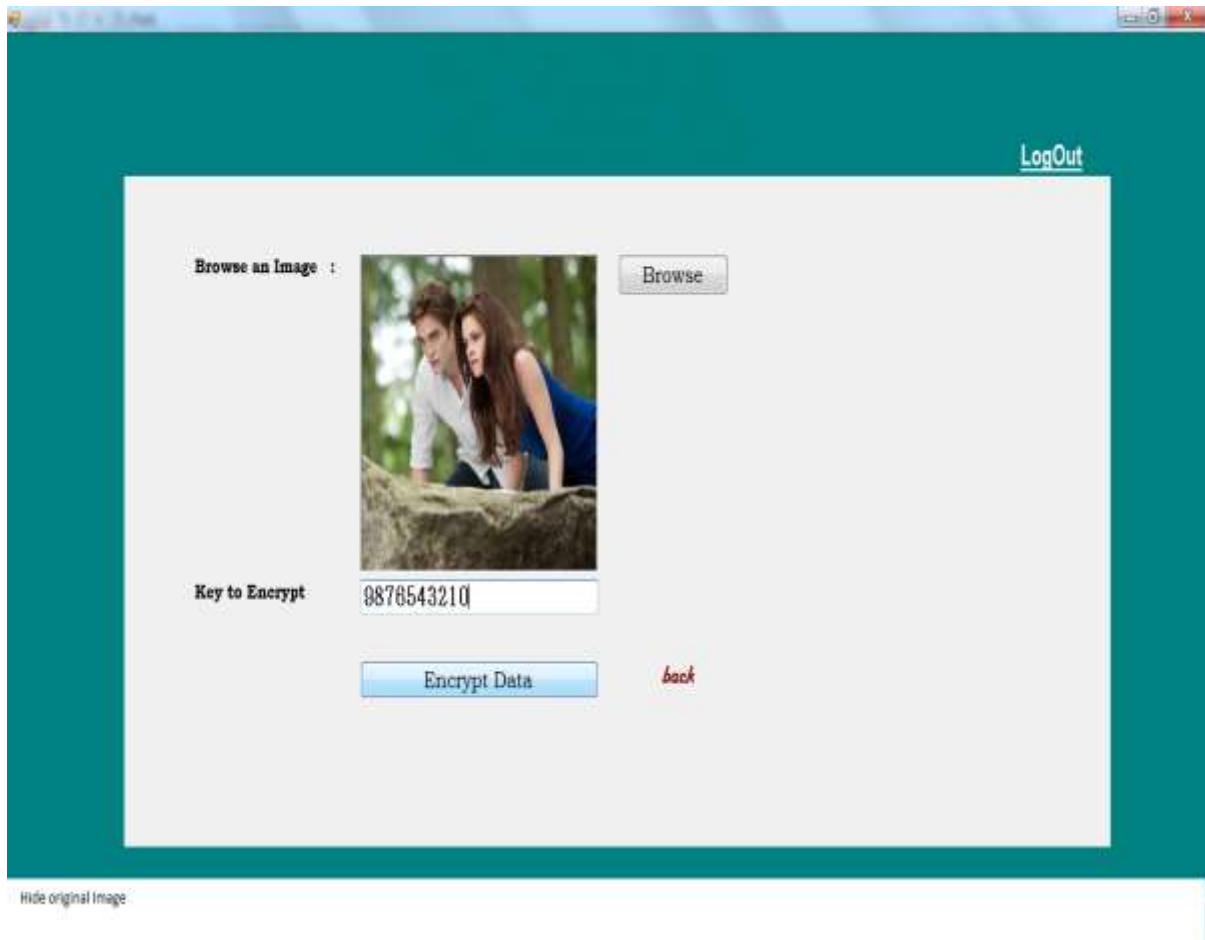
	RECEIVER_ID	RECEIVER_NAME
	9	ramdayal
	9	ramdayal
	9	ramdayal
▶	12	anagha
*		

**back**

When the user fill all the details in the new registration page the data is saved in the database. This table is showing the user Id with the user name. In the first row the user name ramdayal has user Id - 9 similarly in the fourth row the user name anagha has user Id- 12.



Now the user has to click on the create Encryption key. After clicking on the create Encryption key the next page gets open.



In this page the data owner has to select an image in which he wants to hide his secret or confidential data. For this he has to click on the Browse button and need to select that image in which he wish to inject the data. After this he has to enter the key in the key to Encrypt block.

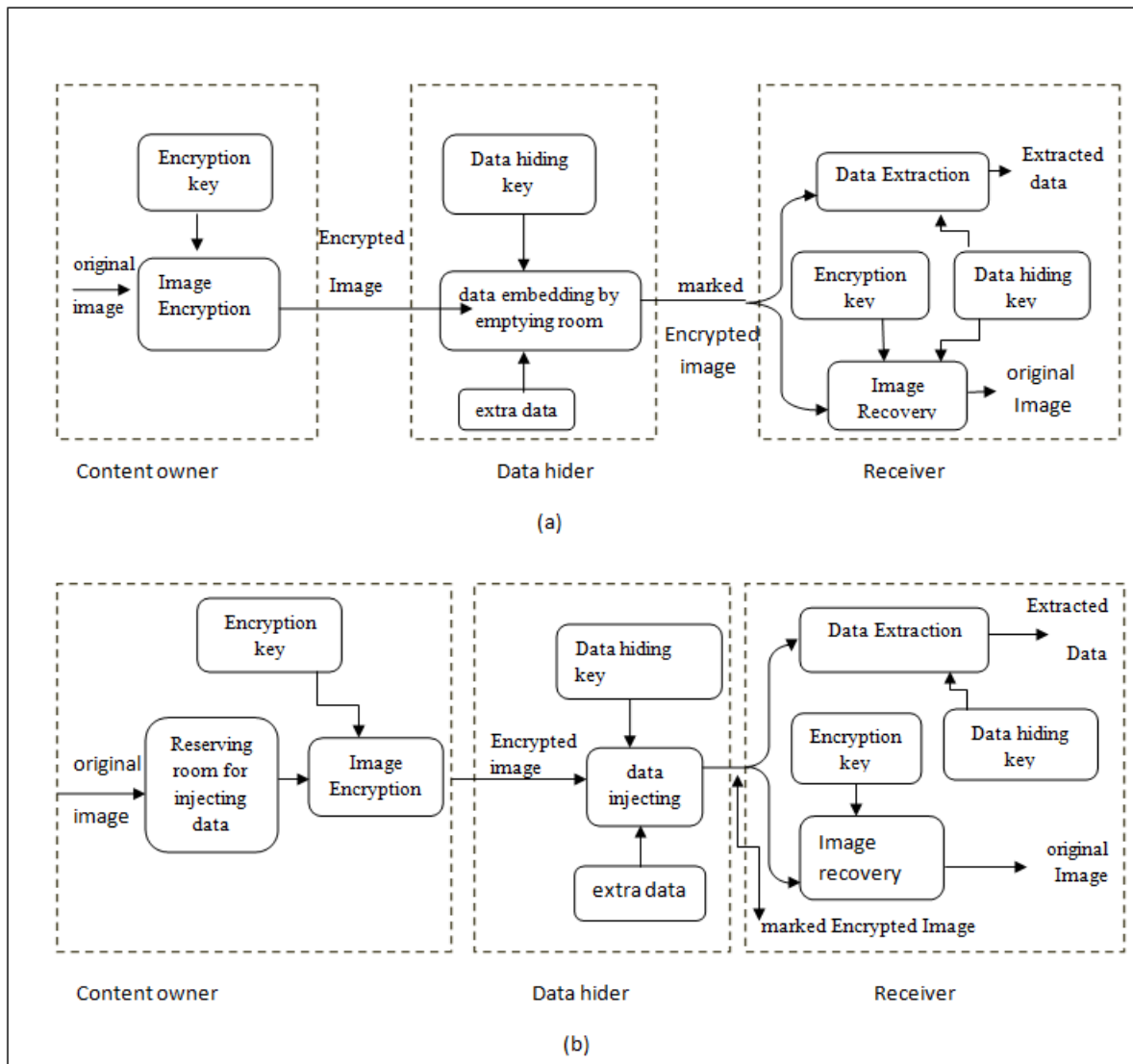
This key is very necessary and it is required to encrypt the image so that it appears in the encrypted form and no one can see that image. The key use to Encrypt an image shall be simple numbers or in alphanumeric form. After this the user has to click on the Encrypt data so that the data he wants to embed in the encrypted image will display in the encrypted form or unreadable form for the security of the confidential data.

Project has three module:-

1. Image encryption
2. Data embedding
3. Image recovery and data extraction.

There are three steps in this experiment.

- i) Encryption
- ii) Data Hiding,
- iii) Decryption and Recovery.



The three steps are as follows:-

**i) Encryption:**

Encryption is the first step. This process contains the image to be encrypted, where we need to select the image to hide our secret information and then enter the alphanumeric encryption key with a special symbol. The output will be an unreadable Encrypted Image. In Encryption image, an image is represented in pixels. Each pixel is represented in using 8 bits.

**ii) Data Hiding:**

Data Hiding is the second step. we have to hide secret data in the encrypted image. In this process, we need a text file which we have to hide inside the encrypted image and an alphanumeric data hiding key. This output of data hiding process is encrypted raw file that encrypts text file into encrypted image.

**iii) Decryption and Recovery**

Decryption and recovery is the third step. Here, if we want to recover the hidden information in the text file together with the image, first we need to recover the information in the specific folder by applying the recovery process to the raw file using the hiding key for the data. Now decrypt Encrypted image with the symmetric key i.e. encryption key from the first step to obtain the original image. The process output will be accurate, original image recovery free of distortion.

#### **IV. Conclusion**

This research provides complete security to confidential data. It presents a new reversible data hiding technique for encrypted image, consisting of three phases through image encryption, data embedding, and data extraction / image recovery. Private data is perfectly protected. It is possible to correctly remove the injected data while the original image can be recovered accurately along with the data hidden inside the distortion-free image. Without knowledge of encryption key and data hiding key, extracting additional data and recovering the original image by third parties is still impossible, which is a more Secure way to send a message across the network.

#### **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.
- [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]. 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.
- [6]. X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
- [7]. 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.
- [8]. X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.
- [9]. 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. 562 IEEE Transactions On Information Forensics And Security, Vol. 8, No. 3, March 2013
- [10]. 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