Reversible Data Hiding For Privacy Preservation in Cloud Data Management

[1] Shruti R. Joshi
[1]ME (IT) Student
[1] Department of Information Technology,
[1]RMD Sinhgad College of Engineering, Pune, India

Abstract* - Encryption is an effective and popular way of converting the original and meaningful content to the cipher form. There is numerous works on data hiding in the encryption field in which reversible data hiding scheme is investigated. Due to reversible property, the original digital content can be completely regained. Reversible data hiding (RDH) with encrypted images is a system by which the original image can be recovered without loss after the embedded message is extracted. RDH for encrypted images by Vacating Room after Encryption (VRAE) is already available, which hides the data by reversibly vacating space from the encrypted images. This existing system has many drawbacks like small payloads embedment, degraded performance in image restoration measured under the standard parameter of PSNR (Peak Signal to Noise Ratio). The proposed method, Reserving Room before Encryption (RRBE) with RDH algorithm for embedment and blowfish algorithm for encryption improves the quality of image restoration and enhances the payload embedding capacity. In the proposed method, content owner uploads the cover image and text to be embedded into it on the cloud space. The RRBE on cloud first reserves enough space by vacating room for embedding data. Next, it converts the image into its encrypted form with the help of encryption key. Then the secret data taken as input is embedded into the space from image on cloud. The encrypted image with secret data is stored in cloud platform. The legitimate receiver then extracts secret data and original image by decryption with preserving privacy and lossless image restoration. The RRBE method applied in cloud data management will satisfy the privacy preserving need of cloud environment efficiently. In this case, PaaS (Platform as a Service) model of a cloud technology is implemented. Also, blowfish algorithm provides strong encryption than simple stream cipher in the existing systems.

***Index Terms* - Reversible Data Hiding, Image Encryption, Privacy preservation, Cloud data.**

I. INTRODUCTION

In the recent era of computer technology and social computing, a large number of databases of media are being exponentially created and stored. As the image data contains sensitive information, the security of image data is also a major concern. Therefore it is becoming a challenge to provide privacy protection to this stored image data. This is due to the fact that cloud is an open source operated by any external third party service provider. So that it is most important that security must be provided in the image service outsourcing design at the time of image transformation and storage. So that owner's data can be well protected.

Nowadays, various techniques such as the steganography are used. It is a process of concealing a message, confidential data or file within another image or file. Watermarking and digital steganography are the two types of data embedding technologies to obtain hidden communication and authentication. The objective of steganography is to conceal a secret message inside harmless medium in such a way that it is not easily revealed to identify that there is a secret message hidden inside. The medium for data hiding is termed as carrier, cover or host.

Since few years, the safety of multimedia data is seen as very important issue. Most multimedia data embedding techniques are being modified. These techniques involve the distortion of host signal in order to insert the confidential message. Generally, this embedding distortion is small, but irreversible [1]. It cannot be extracted to recover the original host signal i.e. image in most of the cases. In many applications, the loss of host signal strength is not prohibitive as long as original and modified signals are perceptually equivalent. The security of the multimedia data can be achieved with data hiding algorithms or encryption. The data compression comes into the picture to decrease the transmission time. The proposed system is working on the problem to combine the encryption, data hiding and compression into the single step. Few solutions have been proposed for example, combining the image encryption and compression. A new challenge was to embed data in encrypted images. However, in a number of areas such as legal and medical imaging, military permanent loss of signal strength is undesirable [1]. This signifies the need towards reversible (lossless) data hiding methods. The majority of the work on reversible data hiding highlights the data embedding/extracting on the plain spatial domain.

II. LITERATURE REVIEW

Reversible steganography can restore the original carrier without any distortion or with ignorable distortion after the extraction of confidential data. So reversible data hiding is now getting popular.

Different Approaches

RDH: Reversible data hiding (RDH) is a technique, using which the original image can be recovered without loss after the extraction of embedded message. RDH has attracted considerable research interest recently. Reversible data hiding facilitates huge possibility of applications to link two types of data in such a way that the cover image can be recovered without loss after the hidden data is extracted out. This provides an additional opportunity of handling two different sets of data [5]. There are many RDH techniques have been worked nowadays. A general framework for RDH was constructed by Fridrich et al. [3]. Compressible features of original image are extracted first and then are compressed them without loss. In this way, vacant space is obtained to store the auxiliary data. The computational complexity of novel RDH technique is low and the time of execution is short. Its overall performance is improved compared to many existing data hiding algorithms.

Reversible Data Embedding Using a Difference Expansion: The aims of this technique were to obtain very high embedding capacity and keep the distortion low. The DE (Difference Expansion) technique is introduced for this purpose. In this, expansion (like multiplication by 2) is carried out on the difference of each pixel group. Next the least significant bits (LSBs) of difference that are all-zero can be used for embedding messages [4]. The motivation of reversible data embedding is distortion free data embedding. It discovers extra space for storing by utilizing the redundancy in the image content.

Rate-distortion Model for RDH: There must be some parameters to balance the proportion of information embedded in image and distortion caused in the image. Distortion is the side effect of many data hiding schemes that causes in the host signal (i.e. image) due to data embedment. Theoretically, reversible data hiding schemes allows complete and blind distortion of the original host data, which is not the practical case. Recursive code construction is the practical way of using RDH by finding the capacity of embedment into host signal and its corresponding distortion [8].

RDH with histogram shifting: The technique uses the zero or the minimum point of the histogram and then slightly modifies the pixel grayscale or color values to embed data. The technique can be applied to almost all types of images. It has been effectively tested on different image types, such as medical, texture, aerial images, and from CorelDRAW database having all of the 1096 images. The computation of technique is pretty simple and the execution time is fairly short. The proposed lossless data hiding technique can be applied to still images and the videos which consist of a sequence of images [6].

Existing Frameworks

Reversible data hiding techniques can be generally implemented into two frameworks existed so far.

- Vacate room after encryption [VRAE]
- Reserve room before encryption [RRBE]

In the first framework, named as vacate room after encryption, the source image is encrypted by a content owner first using a strong encryption algorithm like AES using an encryption key. After the creation of encrypted image, the content owner transfers it towards a data hider to hide some confidential data into the encrypted image by lossless room vacation technique and using a data hiding key. At the receiver side, the content owner or third party authorized receiver can extract the original data using the data hiding key and retrieve the original image from the encrypted version using the encryption key.The disadvantages/challenges with VRAE were such that:

1. The hackers could recover the embedded data in original image by discovering the data placed in particular bit position.
2. Image restoration was not perfect counted under the standard parameter of PSNR.
3. Loss of image secrecy and degraded performance.
4. The image could embed with smaller size of payloads only.

To overcome all the disadvantages of VRAE framework, the proposed system RRBE embeds data by reserving room before encryption using RDH algorithm called as RRBE (Reserving Room before Encryption). It achieves real reversibility which means extraction of data and recovery image are free from error. The data hider can get the advantage of the extra space emptied out to make data hiding process easy. In RRBE, content owner first reserves enough space by vacating room for embedding data. After that it converts the image into encrypted form with the help of encryption key. The data hider then embeds the secret data into the space vacated by content owner. The encrypted image with secret data is stored in cloud platform. Receiver then extracts secret data and original image by decryption. It assures the privacy and lossless image restoration [2]. The RRBE provides security and privacy for the original image and the confidential data embedded within that image stored in cloud environment [1]. Obviously, compared to VRAE framework, in the RRBE, the redundant image content is compressed without loss and then encrypts to preserve privacy.

III. Methodology

Overall Architecture

The goal of system is security provision and privacy preservation in the cloud data management. Cloud provides the data storage and platform as a service. But security and privacy are major concerns on cloud as cloud is a open environment mostly operated by third party. As the image data sets are intended to be communicated among particular group of user, their confidentiality becomes the critical issue. The original image distortion is not permitted in many critical areas like information forensics and security, medical imagery, military imagery etc. Also the confidential data is to be sent over that image. The Reversible Data Hiding with RRBE technique is to be developed to solve above problem of security provision. Due to privacy preservation quality of the proposed method, it can be applied for security purpose of original images and secret data stored in the cloud environment. User will login to the cloud environment. He then chooses the cover image and the message to be hidden in that image and uploads them as input to RRBE framework on cloud.

The RRBE (Reserving room before encryption) framework on the cloud is shown in Fig. 1. It consists of following stages:

- Encrypted Image Generation
- Data hiding in Encrypted Image
- Extraction of Data
- Image Recovery.

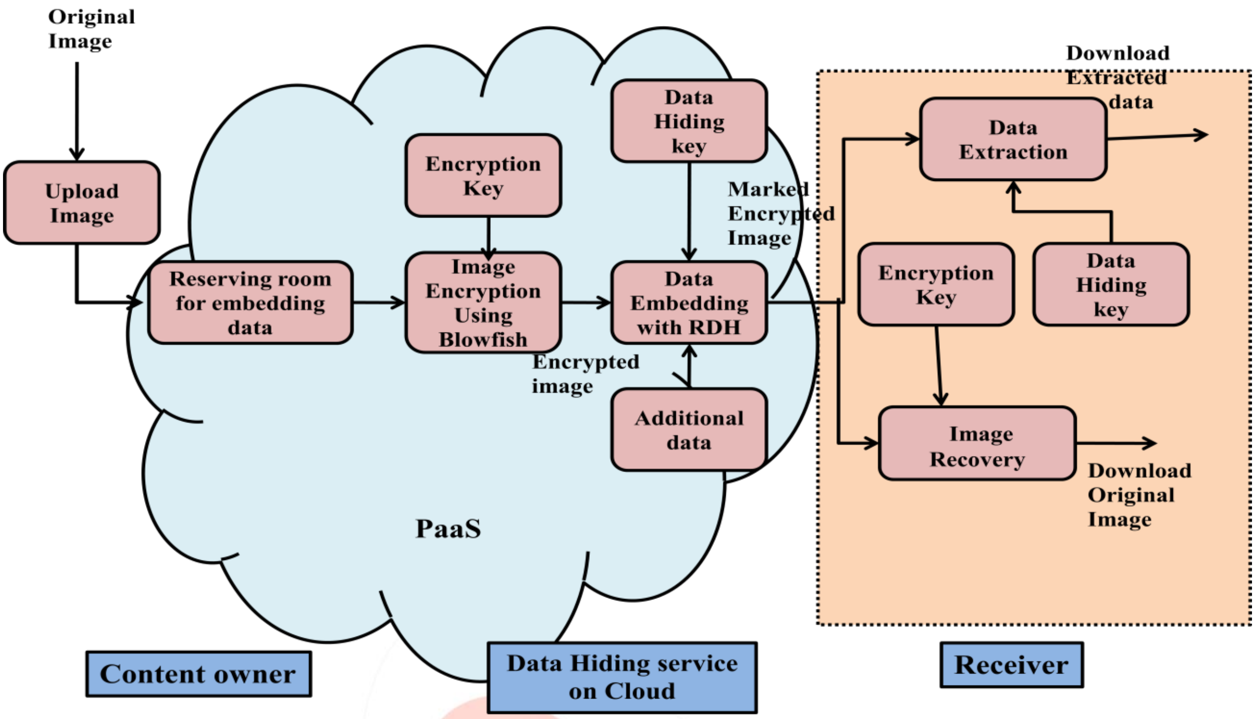

Figure 1.RRBE (Reserving Room Before Encryption) Framework on Cloud

Encrypted Image Generation

The first stage can be split into following three stages: Image Partition, self reversible embedding and image encryption.

Image Partition: Let, C be the input image, A be the complex texture of image and B be the smoother area to be constructed. To construct smoother area(B) is the aim of image partitioning. Block with more complex texture(A) is selected firstly. Then, two or more LSB-planes of A are reversibly embedded into B on which RDH is applied. To identify the smooth area, for each block, first-order smoothness function denoted as *f* is calculated using equation as:

$$f = \sum_{u=2}^{m} \sum_{v=2}^{N-1} \left| C_{u,v} - \frac{C_{u-1,v}+C_{u+1,v}+C_{u,v-1}+C_{u,v+1}}{4} \right| \qquad (1)$$

The original image C is a grayscale or color image with its size $M \times N$ and pixels $C_{i,j}$, $1 \leq i \leq M$, $1 \leq j \leq N$. Size of to-be-embedded messages be *l*, from which the number blocks to be made are decided. Every block should consists of *m* rows such that $m = \lceil {}^{1}/_{N} \rceil$ and *n* be the number of blocks as $n = M - m + 1$. Higher *f* signifies the blocks which contain relatively more intricate textures. The content owner, uses first order smoothness function to select the particular block having highest *f* value and keeps it to the front of the image (part A) concatenated by the rest part B with less complex textured areas. The content owner have the choice to embed two or more LSB-planes of part A into part B[1]. It reduces the size of A to half or more than half. However, the data embedding in the second stage degrades the performance of A, in terms of PSNR, with growing bit-planes exploited.

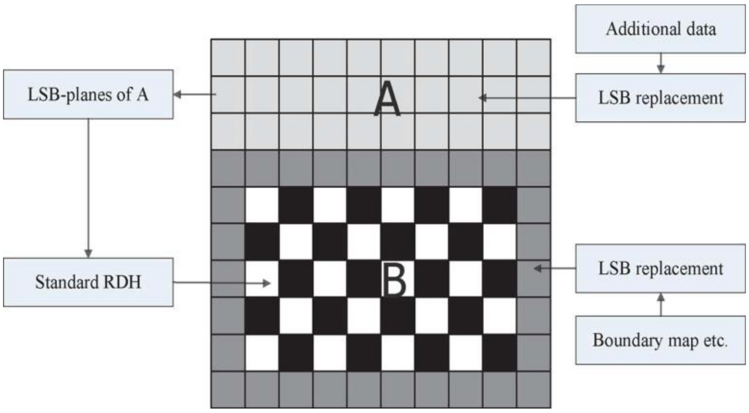

Figure 2.Image Partition and Embedding Process

Self Reversible Embedding: Self embedded image is then encrypted using standard stream cipher and the key. Data hider (third party) can't access the content of original image. Privacy of the content owner remains protected. The objective of self-reversible embedding is to insert the LSB-planes of part A into part B by executing traditional RDH algorithms. The process of self-embedding is described in [6]. Pixels from the part B of image are first classified into two sets: white pixels with indices i and j that satisfy the condition (i+j) mod 2 = 0 For the black pixels also, indices have to meet (i+j) mod 2 = 1. Then, each white pixel, $B_{i,j}$ is obtained by the interpolation value calculated with the four black pixels adjacent to it as follows:

$$B'_{i,j} = w_1 B_{i-1,j} + w_2 B_{i+1,j} + w_3 B_{i,j-1} + w_4 B_{i,j+1} \qquad (2)$$

Here the weight w_i, $1 \leq i \leq 4$, is determined by the same method which is described in [6]. The estimating error is defined using formula $e_{i,j} = B_{i,j} - B'_{i;j}$ .Some data can be added into this estimating error sequence with the help of histogram shift. After that, the estimating errors of black pixels using adjacent white pixels are calculated that may have been modified. Using the histogram shift method, histogram is divided into two parts as left part and right part.

Image Encryption: X, be the self embedded image. In the present step, X is encrypted into E (As shown in Figure 3). The encryption version of X is obtained using the Blowfish algorithm. It takes the blocks of 64 bits and the variable key size long up to 448 bits.

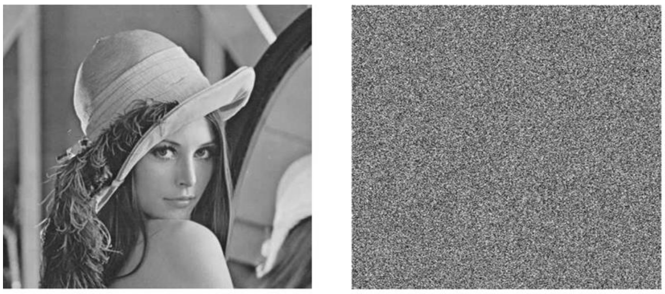

Figure 3.Original image “Lena” and It’s Encrypted version

Data hiding in Encrypted Image

After acquiring the encrypted image, the data hider can embed some data (payload) into it. But, he does not get access to the original image. By knowing the number of bit-planes and rows of pixels, one can able to modify, the data hider simply performs LSB replacement of the available bit-planes with additional data (m). The process of embedment starts with the encrypted version of A, which is denoted by A_E. As A_E has been placed to the top of E, it is easier for the data hider to read 10 bits information in LSBs of first 10 encrypted pixels. Finally, the data hider sets a label to find out the end position of embedding process and encrypts according to the data hiding key to formulate marked encrypted image denoted by E'. Anyone who does not know about the data hiding key, will unable to extract the additional data.

Extraction of Data and Image Recovery

The data extraction process is completely autonomous from image decryption process. The order of their execution can be changed as per the need of application. When extraction is applied as a first process, the encrypted image is remained with database manager. The previously embedded data is extracted out and updated with new data within A_E. The process can be repeated again and again for that encrypted image. This case is applied when the client’s privacy is crucial. In the other case when image is also needed by the user, encryption is performed first and then the extraction of data.

Let, X” be the marked decrypted image that contains the parts as A” and B”. The content owner or intended receiver performs the following steps [7]:

Step 1. Decrypt the image with the help of decryption key provided by blowfish algorithm except the LSB-planes of $A_{E.}$

Step 2. Rearrange A” and B” to obtain plain image with embedded data.

Step 3. Decrypt LSB-Planes from A” using data hiding key. Extract the data up to label.

Step 4. Scan B” part. If no black pixels involved in embedding, go to step 6.

Step 5. Calculate estimating error $e'_{i,j}$ of the black pixels. Recover the estimating error and original pixel value to extract embedded bits. Repeat until the part of payload is extracted.

Step 6. Calculate estimating error $e'_{i,j}$ of the white pixels. Recover error and extract the embedded bits as in step 5. The part B is wholly recovered.

Step 7. Obtain original image C by replacing LSB-planes of A”, with its original bits that are extracted from B”.

Analysis

In the analysis step, comparative analysis of different images against their embedding rate and PSNR is to be done with the help of actual experiments. The standard grayscale or color images like 'Lena', 'Baboon', 'Boat' etc. are used firstly.

$$\text{PSNR}=10 \times \log_{10}\left(\frac{255 \times 255}{MSE}\right) \qquad (5)$$

The security analysis is made by comparing the encryption techniques used which are stream cipher, AES algorithm, DES algorithm with used Blowfish algorithm. The blowfish has better performance than other algorithms in terms of time required and size of file used for encryption.

IV. Experimental Evaluation and Results

The performance of the system is calculated by analyzing the histograms of the original image and the extracted image. Also, the PSNR value is estimated for measuring the quality of the extracted image. It is seen that the PSNR decreases as the embedding rate increases. Also, the PSNR(with same payload and same embedding rate) for grayscale images is greater than the PSNR for color images(same size/dimensions and same images taken),

Embedding Rate Experiment

Table 1 PSNR(dB) comparison under various embedding rates for Standard grayscale Images

	Embedding Rate(bpp)							
	0.005	**0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Images								
Lena	35.23	35.23	32.66	29.79	27.054	25.67	24.42	23.63
Boat	35.56	35.56	32.58	29.99	27.55	26.019	25.091	24.17
Peppers	35.70	35.70	32.24	29.43	26.78	25.29	24.42	24.65
Baboon	40.14	40.14	38.57	36.71	34.81	33.48	32.73	32.47
Barbara	37.63	37.63	35.96	34.24	31.51	29.35	27.99	27.67

As shown in the Table 1, the standard images of size 512 X 512 are given as input to the RRBE framework. The payload is kept constant to observe the change in PSNR values as the embedding rate is increased. It is seen that the PSNR decreases with increase in the embedding rate values. Table 2 shows the same observation if the images used are colored images of same size andthe same payload.

Table 2 PSNR(dB) comparison under various embedding rates for Standard Color Images

	Embedding Rate(bpp)							
	0.005	**0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Images								
Lena	38.35	38.35	35.48	31.95	29.19	27.77	26.92	26.18
Peppers	38.47	38.47	36.11	32.73	29.79	28.10	27.18	26.78
Baboon	41.56	41.56	40.09	38.36	36.93	36.15	35.59	35.13
Barbara	38.80	38.80	36.23	33.65	31.11	29.42	28.10	27.05
Airplane	37.96	37.96	35.57	33.10	31.06	29.93	29.35	28.85

Payload Experiment

Table 3 PSNR(dB) comparison under various Payloads for Standard Images

	Payload(bytes)				
	18	**466**	**1920**	**3350**	**11500**
Images					
Lena	35.23	35.23	35.23	35.23	35.23
Boat	35.56	35.56	35.56	35.56	35.56
Peppers	35.70	35.70	35.70	35.70	35.70
Baboon	40.14	40.14	40.14	40.14	40.14
Barbara	37.63	37.63	37.63	37.63	37.63

Also, in another experiment, same standard grayscale images are embedded with certain payloads. Their performance in terms of PSNR is observed. It is seen that for some range of payload, the PSNR remains constant. When same experiment is performed with the color images, then also constant PSNR is observed.

Encryption Algorithm Experiment

In the encryption phase, both image and the text to be embedded in that image are encrypted. The standard set of images ‘Lena’, ‘Barbara’, ‘Baboon’, ‘Boat’, ‘Peppers’ are encrypted using two encryption algorithms Blowfish and the AES. The experiment is done using same size of payload. It is found that the time required by blowfish is less than the time required by AES for same input. This proved that blowfish is efficient than AES in terms of time. In figure 4. X-axis denotes the names of images while Y-axis denotes the time in milliseconds required by each algorithm for encryption process.

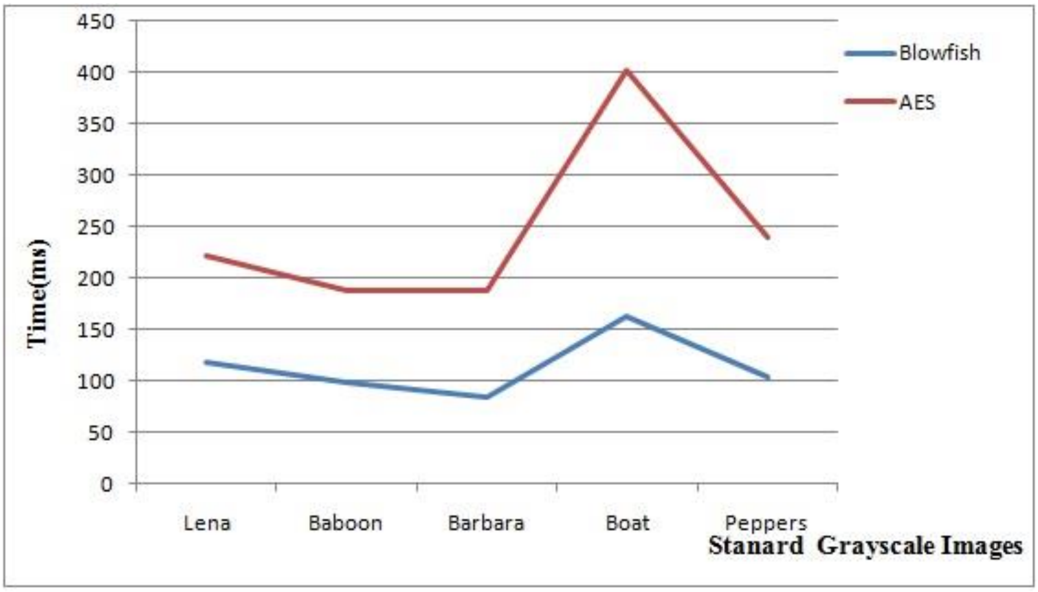

Figure 4.Comparison of Encryption Algorithms Blowfish and AES

V. Conclusion

RDH in encrypted images is a new technology which is drawing enormous attention because of its ability to uphold the content owner's privacy and maintain integrity of data. Also real reversibility of data can be implemented, that is data extraction and image recovery is free from any error. The requirements from cloud data management are same. The proposed system implements RDH in encrypted images by vacating room before encryption, which is exactly opposed to the existing method of RDH, vacating room after encryption. Thus, the data hider gets advantage from the extra space which is created by vacating the room in advance. It makes the data hiding process easy. So the method can take benefit of all previous RDH techniques for plain image and attain extremely good performance without loss of privacy and quality of data on cloud. Deployment of system on cloud environment assures the security and privacy for the images and data stored on it. Also, the efficient and strong encryption of blowfish ensures the better security than previous stream cipher and AES encryption techniques.

VI. Acknowledgment

First and foremost, I would like to express my sincere gratitude to my guide Ms. Prachi Sorte, for her continuous support and Head of the Department Ms. Sweta Kale in completion of my project. I will always be grateful for her patience, motivation, enthusiasm and immense knowledge. This work is based on the research efforts of Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu, and Fenghua Li. I am very much indebted to them for their deep insight and inspiring work which is a boost for future researchers. Finally, I would like to pay my respects and love to my parents, for their love and encouragement throughout my career.

References

[1] K. Ma, W. Zang and X. Zhao, "Reversible Data hiding in Encrypted Images by reserving Room Before Encryption", IEEE Trans. Information Forensics and Security, vol. 8, no. 3, pp. 553-562, Mar. 2013.

[2] 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. 7176.

[3] 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.

[4] J. Tian, "Reversible data embedding using a difference expansion", IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890896, Aug. 2003.

[5] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding", IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354362, Mar. 2006.

[6] 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.

[7] L. Luo et al., "Reversible image watermarking using interpolation technique", IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187-193, Mar. 2010.

[8] 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. 29923006, Oct. 2004.