



Enhancing the Security of the Image using Reversible Image Transformation

¹T Suguna, ²C Aswini, ³S Gladson Oliver and ⁴R Malavika

¹Assistant Professor, ²Assistant Professor, ³Assistant Professor and ⁴Assistant Professor

¹Department of Information Technology,

¹Government College of Technology, Coimbatore, India

Abstract : In this global world, security is one of the essential features of any application. Every data should be protected in one way or other. Images get hacked and manipulated by intruders or hackers. The confidential images should be protected from being attacked. Encrypting the images protects its content. But the hackers may try to decrypt it. So in order to avoid the hackers from attacking the images, the image can be transformed into another image. So in this project, Reversible Image Transformation technique is used to transform an image into another image. The contents required to transform is encrypted and embedded into the image using Difference Expansion technique. The image after entire processing would resemble the target image used. Since Difference Expansion is a Reversible Data Hiding technique, the data embedded can be retrieved back. The data retrieved is decrypted and used to transform the image back to its original form. The original image is retrieved back with minimal loss in quality. The payload that can be embedded into the image using Difference Expansion is very large when compared to other data hiding scheme such as LSB Embedding. Moreover the quality of the image is also preserved.

IndexTerms – Image Transform, Data Embedding, security, difference expansion.

I. INTRODUCTION

Nowadays most of the content, including multimedia, become online. The data is transferred across the Internet daily. Huge amount of images and videos are uploaded and downloaded from the Internet. The privacy of the image contents have to be protected. It is a challenging task to protect the images from being attacked. The contents of the images should also be preserved in a lossless way.

So far, the images have been converted into an encrypted form, which attracts the hackers to try to decrypt the images to get contents of it. The contents of the image might get lost during the conversion. But in this paper we propose a technique which transforms the image into another image. Since this technique is reversible, the original image can be retrieved without loss in quality of the image. The information required to transform the image have to be hidden in the image. Moreover the hidden information should also be retrieved without any loss.

Many RDH methods on images have been proposed. In essence, all these methods can be viewed as a process of semantic lossless compression in which some space is saved for embedding extra data by losslessly compressing the image. From the viewpoint of compression, these methods on RDH-EI belong to the two frameworks: Framework I “vacating room after encryption (VRAE)” and Framework II “reserving room before encryption (RRBE).”

In the framework ‘VRAE,’ the cloud server embeds data by losslessly vacating room from the encrypted images by using the idea of compressing encrypted images. Compression of encrypted data can be formulated as source coding with side information at the decoder. In the framework “RRBE,” the image owner first empties out room by using RDH method in the plain images. After that, the image is encrypted and outsourced to the cloud and the cloud server can freely embed data into the reserved room of the encrypted image. The first method under RRBE framework which reserves room by embedding LSBs of some pixels into other pixels with a traditional RDH method and then encrypts the image, so the positions of these LSBs in the encrypted image can be used to embed data.

For both frameworks, VRAE and RRBE, the image owner will send a ciphertext-formed image to the cloud. However, the ciphertexts with the special form of messy codes are easy to cause the attention of the cloud server who may try to dig out information on the encryption users. In fact, the cloud server is assumed to be curious to collect information from the out-sourced files and obviously the encrypted images are more attractive to a curious cloud server. Therefore, the fact, that the user is outsourcing encrypted images, itself is also a kind of privacy of the user, which should be protected.

In this paper, we propose a novel framework for RDH-EI by using reversible image transformation (RIT). RIT transfers the semantic (content) of the original image I into the semantic of another image J , and “reversibility” means that I can be losslessly restored from the transformed image. Therefore RIT can be viewed as a special encryption scheme, called “Semantic Transfer Encryption.” In other words, the resultant transformed image which is also the encrypted image $E(I)$ will look similar with J . The image J is selected to be irrelevant with I but has the same size of I , and thus the content of the image I is protected. The contents required to transform are hidden using Difference Expansion technique.

The rest of the paper is organized as follows. A method of RIT is elaborated in Section II. Difference Expansion technique is described in Section III. The paper is concluded with a discussion in Section IV.

II. AN EXAMPLE ON REVERSIBLE IMAGE TRANSFORMATION

In this section, we propose a method of RIT to encrypt spatial images, which is inspired by the technique of image transformation proposed by Lee and Tsai [25]. Lee et al.’s method can transform the original image to a freely-selected target image with the same size, yielding a secret-fragment-visible mosaic image defined in [24]. But the original image cannot be restored in a lossless way. It is not reversible, so it is not suitable for the scenario of RDH-EI. We will modify Lee et al.’s method to be reversible and obtain an encrypted image which looks like the target image.

For color images, we transform the color channel R, G, and B respectively in the same manner. So we just take gray images (one channel) as an example to describe the method. For an original image I , we randomly select a target image J having the same size with I from an image database.

Firstly, we divide the original image I and the target image J into N non-overlapping blocks respectively, and then pair the blocks of I and J as a sequence such that $(B_1, T_1), \dots, (B_N, T_N)$, where B_i is an original block of I and T_i is the corresponding target block of J , $1 \leq i \leq N$. We will transform B_i toward T_i and generate a T_i similar to T_i . After that, we replace each T_i with T_i in the target image J to get the transformed image J . Finally we embed some accessorial information (AI) into J with an RDH method and generate the ultimate “encrypted image” $E(I)$. These AI is necessary for recovering I from J . Before being embedded, these AI will be compressed and encrypted with a key K shared with the receiver, so only a receiver having K can decrypt $E(I)$. The proposed transformation process consists of three steps: block pairing, block transformation and AI embedding. We will mainly elaborate the first two steps in the sections and the third step is described in Section III.

A. Block Pairing

To make the transformed image J look like target image J , we hope, after transformation, each transformed block will have close mean and standard deviation (SD) with the target block. So we first compute the mean and SD of each block of I and J respectively. Let a block B be a set of pixels such that $B = \{p_1, p_2, \dots, p_n\}$, and then the mean and SD of this block is calculated as follows:

$$u = \frac{1}{n} \sum_{i=1}^n p_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - u)^2}. \quad (2)$$

When matching blocks between original image and target image, we hope two blocks with closest SDs to be a pair. In Lee et al.’s method, the blocks of original image and target image are sorted in ascending order according to their SDs respectively, and then each original block is paired up with a corresponding target tile in turn according to the order. To recover the original image from the transformed image, the positions of the original blocks should be recorded and embedded into the transformed image with an RDH method. If the image is divided into N blocks, $N \log N$ bits are needed to record block indexes. Obviously, the smaller the block size is, the better the quality of transformed image will be, but which will result in a large N . Therefore, the amount of information used to record the index for each block may be so large that it will cause much distortion when embedding these information into the transformed image. In fact there may not exist enough redundant space to store these additional information.

To compress the block indexes, we first classify the blocks according to their SD values before pairing them up. In fact, we found that the SD values of most blocks concentrate in a small range close to zero and the frequency quickly drops down with the increase of the SD value. Therefore, we divide the blocks into two classes with unequal proportions: class 0 for blocks with smaller SDs, and class 1 for blocks with larger SDs, and pair up the blocks belonging to the same class. By assigning the majority of blocks to the class 0, we can avoid the large deviation of SDs between a pair of blocks and efficiently compress the indexes at the same time.

In this paper, we propose to divide both the original and target images into non-overlapping 4×4 blocks and calculate the SDs of each block. We first divide the blocks of original image I into 2 classes according to the quantile of SDs. Denote that the α quantile of SDs by $N\alpha$. We assign the blocks with SDs $\in [0, N\alpha]$ to “Class 0,” and blocks with SDs $\in (N\alpha, N]$ to “Class 1.” And then we will scan the blocks in the raster order, i.e., from left to right and from top to bottom, and assign a class label, 0 or 1, to each block. Next, we label the blocks of target image based on the classes’ volumes of original image. Assuming that the i th class in the original image includes n_i blocks for $i = 0$ or 1 , we scan the target image in the raster order, and label the first n_0 blocks with the smallest SDs as Class 0, and the rest n_1 blocks as Class 1. As a result, each class in the target image includes the same number of blocks as the corresponding class in the original image. We pair the original block up with target block in the following manner. Scan the original image and target image in raster order respectively and pair the j th block of the class i in the original image up with the

j th block of the class I in the target image for $i = 0, 1$ and $j = 1, \dots, n_i$. A simple example on the proposed block pairing method is shown in Fig. 3, in which the image only consists of 10 blocks. By setting $\alpha = 70$, we assign 7 blocks with smallest SDs into class 0, and the rest 3 blocks into class 1 in the original image. In the target image, although the 8th and 9th block have the same SD value 5, the 8th block is assigned to class 0 but the 9th block is assigned to the class 1, because class 0 can only include 7 blocks as determined by the class 0 of the original image. After labeling the class indexes, we get a class index table (CIT) for original image and target image respectively, which will be helpful for understanding the procedure of block pairing.

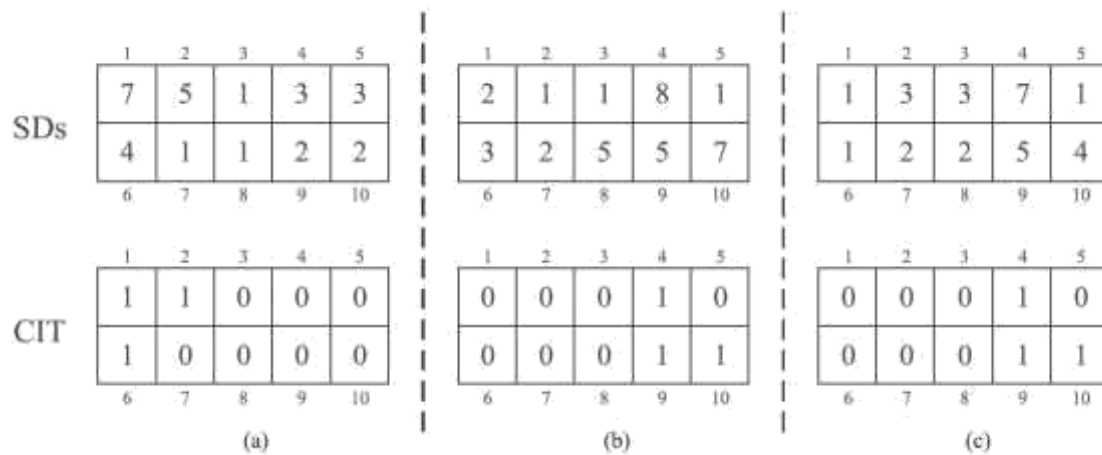


Fig. 1. Example of block pairing. (a) Original image. (b) Target image. (c) Transformed image.

Table I
Block pairing result of the example in fig. 1

Block index of original image	1	2	3	4	5	6	7	8	9	10
Block index of target image	4	9	1	2	3	10	5	6	7	8

According to the pairing rule, the first block of the original image is paired up with the forth block of the target image, because both of them is the first block of class 1 as shown in the CIT; the second block of original image is paired up with the ninth block of target image, because both of them is the second block of class 1, and so on. The pairing result is listed in Table I, which can be generated according to the CIT of original image and the CIT of the target image.

For each pair of blocks (B , T), as we will see in the next section, the original block B will be transformed to target block T by mean shifting and block rotation, yielding T . By replacing each T with T in the target image, the sender will generate the transformed image. Note that both operations of mean shifting and block rotation will not change the SD value, so T has the same SD as B . Therefore, the SDs in transformed image is only a permutation of those in original image. When classifying the blocks of transformed image according to α quantile of SDs, the receiver can get a CIT that is same with the CIT of target image as shown in Fig. 1(b) and (c).

Therefore, to restore the original image from the transformed image, the receiver only needs to know the CIT of the original image. In fact, by CIT of original image and the CIT of transformed image (which is also the CIT of target image), the receiver can reconstruct Table I perfectly. Then according to the table he will know how to rearrange the transformed blocks to restore the original blocks. In the example of Fig. 1, the first block of the transformed image should be put back to position 3, and the second block should be put back to position 4 as indicated in Table I.

B. Block Transformation

By the block pairing method described above, in each pair (B , T), the two blocks have close SD values. Therefore, when transforming B towards T , we only need a mean shifting transformation that is reversible. However, the transformation used in Lee and Tsai's method [25] is not reversible because it changes the mean and SD at the same time. Let the original block $B = \{p_1, p_2, \dots, p_n\}$, and the corresponding target block $T = \{p_1, p_2, \dots, p_n\}$. With Eq. (1), we calculate the means of B and T and denote them by u_B and u_T respectively. The transformed block $T = \{p_1, p_2, \dots, p_n\}$ is generated by the mean shifting as follows:

$$p_i = p_i + u_T - u_B \quad (3)$$

where $(u_T - u_B)$ is the difference between the means of target block and original block. We want to shift each pixel value of original block by amplitude $(u_T - u_B)$ and thus the transformed block has the same mean with the corresponding target block. However, because the pixel value p_i should be an integer, to keep the transformation reversible, we round the difference to be the closest integer as

$$u = \text{round}(u_T - u_B) \quad (4)$$

and shift the pixel value by u , namely, each p_i is gotten by

$$p_i = p_i + u. \quad (5)$$

Note that the pixel value p_i should be an integer between 0 and 255, so the transformation (5) may result in some overflow/underflow pixel values. To avoid such transformed blocks abstained by (5), we assume that the maximum overflow pixel value is OV_{max} for $u \geq 0$ or the minimum underflow pixel value is $U_{Nm in}$ for $u < 0$. If overflow/underflow occurs in some blocks, we eliminate them by modifying u as,

$$u = \begin{cases} u + 255 - OV_{max}, & \text{if } u \geq 0 \\ u - U_{Nm in}, & \text{if } u < 0 \end{cases} \quad (6)$$

We use the modified u to shift the pixels of block B , and thus all the pixels' values are controlled into the range of $[0, 255]$.

However the range of u 's value is still very large, which cannot be efficiently compressed. Thus we further modify u as

$$u = \begin{cases} \lambda \times \text{round} \left(\frac{u}{\lambda} \right), & \text{If } u \geq 0 \\ \lambda \times \text{floor} \left(\frac{u}{\lambda} + \frac{\lambda}{2} \right), & \text{if } u < 0 \end{cases} \quad (7)$$

in which the quantization step, λ , is an even parameter. Then it just needs to record $u = 2 \lfloor u/\lambda \rfloor$, by which it has the advantage of not to record the sign of u . Because when u is an even number it means $u \geq 0$ and when u is an odd number it means $u < 0$. Since when λ is large the amount of information recording u will be small but the offset between the modified u and the original u will be large, a tradeoff must be made by choosing λ . We set $\lambda = 8$ in the following experiments.

After shifting transformation we get a new block T . With these new blocks, we replace the corresponding blocks in the target image and generate the transformed image J . The parameters, u and rotation directions, will be compressed, encrypted and then embedded into the transformed image J as AI to output the "encrypted image" $E(I)$ called in this paper image. The transform and anti-transformation procedures of the pro-posed method are described in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1: Procedure of Transformation.

Input: An original image I

Output: The encrypted image $E(I)$.

- 1) Select a target image J having the same size as I from an image database.
- 2) Divide both I and J into several non-overlapping 4×4 blocks. Assuming that each image consists of N blocks, calculate the mean and SD of each block.
- 3) Classify the blocks with α quantile of SDs and generate CITs for I and J respectively. Pair up blocks of I with blocks of J according to the CITs as described in Section II-A.
- 4) For each block pair (B_i, T_i) ($1 \leq i \leq N$), compute the mean difference u_i . Add u_i to each pixel of B_i which yields a transformed block T_i .
- 5) In the target image J , replace each block T_i with the corresponding transformed block T_i for $1 \leq i \leq N$ and generate the transformed image J .
- 6) Collect u_i 's for all block pairs, and compress them together with the CIT of I .
- 7) Take the encrypted sequence as AI, and embed AI into the transformed image J with an Difference Expansion method, and output the encrypted image $E(I)$.

Algorithm 2: Procedure of Anti-transformation.

Input: The encrypted image $E(I)$

Output: The original image I .

- 1) Extract AI and restore the transformed image J from $E(I)$ with the Difference Expansion.
- 2) Decrypt AI to obtain CIT of I , u_i ($1 \leq i \leq N$) and α .
- 3) Divide J into non-overlapping N blocks with size of 4×4 . Calculate the SDs of blocks, and then generate the CIT of J according to the α quantile of SDs.
- 4) According to the CITs of J and I , rearrange the blocks of J as described in Section II-A.
- 5) For each block T_i of J for $1 \leq i \leq N$, subtract u_i from each pixel of T_i , and finally output the original image I .

III. DIFFERENCE EXPANSION

IV. The DE embedding technique involves pairing the pixels of the host image and transforming them into a low-pass image containing the integer averages and a high-pass image containing the pixel differences. If x and y be the intensity values of a pixel-pair, then l and h are defined as

$$l = \frac{(x+y)}{2}, \quad h = x - y \quad (8)$$

This transformation is invertible, so that the gray levels x and y can be computed from l and h

$$x = l + \frac{h+1}{2}, \text{ and } y = l - \frac{h}{2} \quad (9)$$

An information bit $b \in \{0,1\}$ is embedded by appending it to the LSB of the h difference, thus creating a new LSB. The watermarked difference is

$$h' = 2h + b \quad (10)$$

Figure 2 shows an example of embedding one bit in a pair of pixels. The resulting pixel gray-levels are calculated from the difference (h') and integer average l using equation (9). For an image with n -bit pixel representation, the gray levels satisfy $x, y \in [0, 2^n - 1]$ if and only if h and l

$$h \in \text{Rd}(l) = [0, \min(2(2^n - 1 - l), 2l + 1)] \quad (11)$$

where R is called the invertible region. Combining (10) and (11), we obtain the condition for a difference h to undergo DE

$$2h + i \in R(l) \text{ for } i = 0, 1 \quad (12)$$

This condition is called the expandability condition for DE. A difference that satisfies the expandability condition is called an expandable difference. Meanwhile, the LSB of a difference may be replaced with an information bit. This is a lossy embedding technique since the true LSB is overwritten in the embedding process. However, in Tian's scheme, the true LSBs of the differences that are embedded by LSB-replacement are saved and embedded with the payload, to ensure lossless reconstruction. The LSB of a difference can be flipped without affecting its ability to invert back to the pixel domain if and only if

$$2h/2 + b \in \text{Rd}(l) \text{ for } i = 0, 1 \quad (13)$$

This is called the changeability condition. A difference satisfying the changeability condition is called a changeable difference. An expandable difference is also a changeable difference. An information bit b is embedded into a changeable difference as

$$h' = 2h/2 + b \quad (14)$$

During embedding, differences are classified into three groups: expandable, changeable, and non-changeable. Data bits are embedded only into expandable and changeable differences using equation (10) and (14) respectively. A location map is formed to distinguish between the three different groups. The map is then compressed, concatenated with the payload, and then embedded into the image.

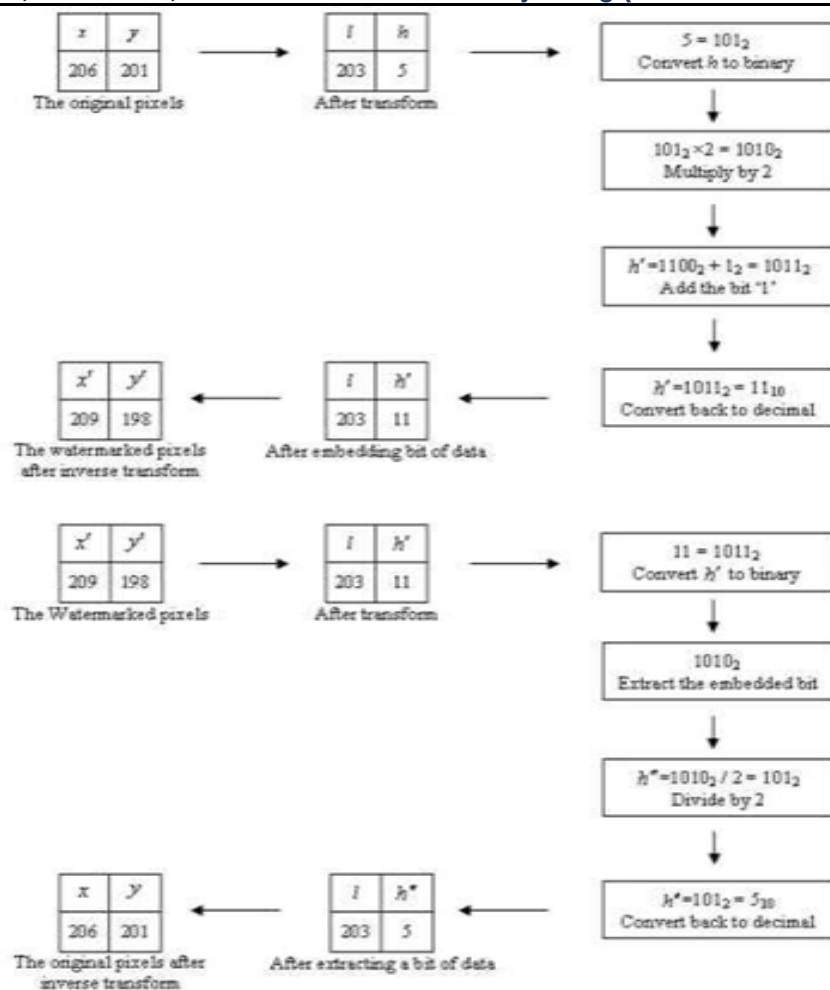


Fig 2. Embedding and extracting phases for a pair

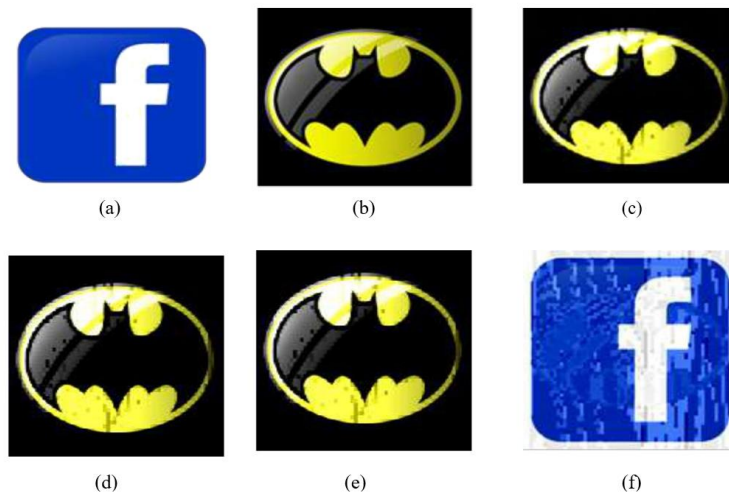


Fig. 3. Experiment results of test images. (a) Original image. (b) Target image. (c) Encrypted image. (d) Watermarked image. (e) Retrieved image. (f) Final image.

V. CONCLUSION AND FUTURE WORK

In this paper, a Reversible Image Transformation technique is proposed. In the proposed scheme, an image is successfully transformed into another image. The data required for the transformation is hidden in the image using Difference Expansion technique. The hidden data is retrieved back successfully.

The retrieved data is used to retrieve the original image from the transformed one with minimal loss in quality. The proposed scheme has an advantage of using Difference Expansion for hiding the data, due to which a large amount (in bits) of payload can be embedded into the image. All the experimental results clearly show that the proposed transformation technique is secure and the original image can be retrieved back with minimal loss in quality of the image. To show the efficiency of proposed work it is compared with a well-known data hiding scheme on images and experimental results show better performance with improved image quality.

The idea of Reversible Image Transformation can be further extended to audio and video. The measures can be taken to improve the quality of the encrypted image. The payload embedded into the image can also be increased without affecting the quality of the image.

REFERENCES

- [1] Weiming Zhan, Hui Wang, Dongdong Hou, Nenghai Yu, 2016, "Reversible Data Hiding in Encrypted Images by Reversible Image Transformation" IEEE Transactions on Multimedia., vol. 18, no. 18.
- [2] X. Zhang, Apr. 2011, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258.
- [3] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, Mar. 2013, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 553–562.
- [4] I.-C. Dragoi and D. Coltuc, Apr. 2014, "Local-prediction-based difference expansion reversible watermarking," IEEE Trans. Image Process., vol. 23, no. 4, pp. 1779–1790.
- [5] R. Rad, K. Wong, and J. Guo, Apr. 2014, "An unified data embedding and scrambling method," IEEE Trans. Image Process., vol. 23, no. 4, pp. 1463–1475.
- [6] J. Fridrich, M. Long, Apr. 2000, "Steganalysis of LSB encoding in color images," IEEE Conf. on Multimedia., vol.3, pp. 1279-1282.
- [7] K. Hwang and D. Li, Sep. /Oct. 2010, "Trusted cloud computing with secure resources and data coloring," IEEE Internet Comput., vol. 14, no. 5, pp. 14–22.
- [8] F. Bao, R. H. Deng, B. C. Ooi, and Y. Yang, Dec. 2005, "Tailored reversible watermarking schemes for authentication of electronic clinical atlas," IEEE Trans. Inf. Technol. Biomed., vol. 9, no. 4, pp. 554–563.
- [9] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, Jul. 2009, "Reversible watermarking algorithm using sorting and prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999.
- [10] B.ou, X. Li, Y. Zhao, R. Ni, and Y. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," IEEE Trans. Image Process., vol. 22, no. 12, pp. 5010–5021, Dec. 2013.
- [11] J. Tian, Aug. 2003, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896.
- [12] X. Hu, W. Zhang, X. Li, and N. Yu, Mar. 2015, "Minimum rate prediction and optimized histograms modification for reversible data hiding," IEEE Trans. Inf. Forensics Security, vol. 10, no. 3, pp. 653–664.
- [13] X. Hu et al. , May 2013, "Fast estimation of optimal marked-signal distribution for reversible data hiding," IEEE Trans. Inf. Forensics Security, vol. 8, no. 5, pp. 779–788.
- [14] W. Zhang, X. Hu, and N. Yu, Jan. 2015, "Optimal transition probability of reversible data hiding for general distortion metrics and its applications," IEEE Trans. Image Process., vol. 24, no. 1, pp. 294–304.