A Flexible Reversible Data Hiding Method in Compressible Encrypted Images

Ryota MOTOMURA*, Shoko IMAIZUMI† and Hitoshi KIYA‡
* Graduate School of Science and Engineering, Chiba University, Chiba, Japan
E-mail: ryo.0098@chiba-u.jp
† Graduate School of Engineering, Chiba University, Chiba, Japan
E-mail: imaizumi@chiba-u.jp
‡ Faculty of System Design, Tokyo Metropolitan University, Tokyo, Japan
E-mail: kiya@tmu.ac.jp

***Abstract*—In this paper, we propose a reversible data hiding method in encrypted images, where both the compression efficiency and hiding capacity are flexibly controlled. The proposed method first divides the original image into two regions, e.g., region of interest (ROI) and non-region of interest (non-ROI). In one region, an encryption-then-compression (EtC) system is used for encryption and a histogram shifting based method is adopted for data hiding. Accordingly, lossless compression using image coding standards is effective in this region even after encryption, and also the region can be decrypted without data extraction. In the meanwhile, an MSB prediction based method is introduced in another region. The MSB prediction errors are first detected to recover the original MSBs in the restoration process. Then, exclusive-or operation and MSB replacement are conducted for encryption and data hiding, respectively. This method attains high hiding rate; the rate in this region is around 1 bpp. Experimental results show the effectiveness of the proposed method in terms of hiding capacity and lossless compression efficiency.**

I. Introduction

Data hiding techniques have attracted attentions over recent decades in the image security field. The main purpose of data hiding is copyright protection by embedding some information into a target medium imperceptibly. In particular, reversible data hiding (RDH) can perfectly retrieve the original image by extracting the embedded payload. It is effective not only for natural images but also for medical, military, evidential images, and so forth. Numerous RDH techniques have been proposed [1]–[6]. Additionally, RDH in encrypted images (RDH-EI) has been actively studied in recent years [7]–[13]. In RDH-EI, it is expected that an image owner first encrypts the original image, and a third party, such as a system or channel administrator, may embed additional information into the encrypted images. In contrast, an image user would desire to obtain the original or high-quality image after decryption even when the restored image still contains the additional information.

A traditional RDH-EI method [8] first divides an image into two regions in spatial domain, and then embeds LSB values in one region into another region. The LSBs in the former region are referred to as the embeddable area. Subsequently, the entire image is encrypted by the exclusive-or operation. A data hider embeds the payload bits into the encrypted embeddable area by LSB substitution. In this method, the original image can be restored after decryption without data extraction, and the maximum hiding capacity is about 0.5 bpp.

Puteaux et al. attained a high capacity method by introducing MSB prediction and substitution for data hiding [9]. Further, Dragoi et al. [10] enhanced the security of this method. However, there still exist multiple cases where the reversibility is not fully ensured. Hirasawa et al. [11] extended Puteaux et al.'s method [9] to guarantee the full reversibility by defining the precise conditions. The hiding capacity is quite high, namely close to 1 bpp, and the mathematical complexity is low. Meanwhile, these methods cannot compress the output images by using image coding standards because the pixel-based exclusive-or operation is adopted for encryption. Hereafter, the method [11] is referred to as the RDH-MSB method.

An RDH-EI method has been proposed with effective compressibility for output images [12]. This method introduces an Encryption-then-Compression (EtC) system [14], [15], and thus can losslessly compress the output image using image coding standards, such as JPEG-LS [17] and JPEG 2000 [18]. Additionally, an image histogram is not transformed before/after the encryption processes. We can thereby embed and extract the payload in plain and/or encrypted domains flexibly by using the RDH method based on histogram shift (RDH-HS) [2]. Further, the marked image quality is significantly high. This method, however, has extremely low hiding capacity, i.e., about 0.1 bpp. Hereafter, we call this method as the RDH-EtC method.

In this paper, we propose a new framework of RDH-EI, which can control both compression efficiency and hiding capacity. Our method first classifies an original image into two regions. In one region, the RDH-EtC method is conducted. Here, the encryption process in the original RDH-EtC method is replaced with Chuman et al.'s method [16], where the conventional EtC systems [14], [15] have been extended to enhance the security. In contrast, the RDH-MSB method is employed in another region. Consequently, the former region can be losslessly compressed by using image coding standards. In the latter region, the hiding capacity is significantly high. Thus, our method can control both the compression

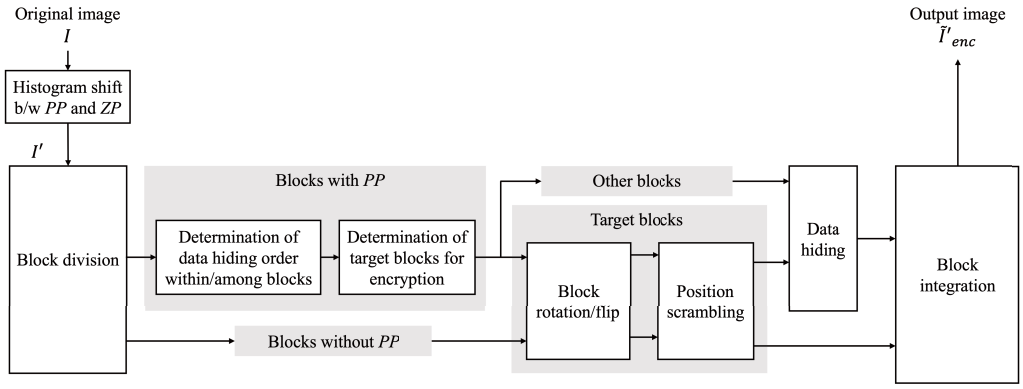


Fig. 1 Block diagram of RDH-EtC method [12].

performance and hiding capacity according to the area ratio of two regions. Through our experiments, we confirm the effectiveness of the proposed method from the aspects of the lossless compression performance using JPEG-LS and JPEG 2000 and hiding capacity.

II. Preparation

A. RDH Method for EtC Images

In the RDH-EtC method [12], the compression efficiency of the output images is well considered. This method uses two processes of the block scrambling based encryption for EtC systems [14], [15] for encryption, namely, position scrambling and block rotation/flip. Since these two scrambling processes do not transform the image histogram, we can flexibly embed and extract the payload by using the RDH-HS method [2]. For instance, even when the payload is embedded after encryption, the output image can be decrypted without extracting the payload. Further, this method can losslessly compress the output images by using image coding standards, such as JPEG-LS [17] and JPEG 2000 [18]. We explain the procedures of the RDH-EtC method in accordance with Fig.1. Here, we assume that the size of the original image I is $M \times N$ pixels.

Step1-1: Explore a pair of the peak and lowest points (hereafter, PP and ZP) , which are the bins with the highest and lowest frequency of appearance, from the histogram of the original image I.

Step1-2: Shift the values of pixels between PP and ZP by +1 or -1. Accordingly, the intermediate image I', where the adjacent bin of PP is empty, is obtained.

Step1-3: Divide I' into multiple blocks with $B_x \times B_y$ pixels.

Step1-4: For the blocks containing PP, determine the data hiding order within/among blocks according to the defined conditions. The blocks without PP are excluded from the data hiding process.

Step1-5: For the blocks containing PP, extract the target blocks for block rotation/flip and position scrambling according to the defined conditions. The blocks without PP are the target blocks for both of the scrambling processes.

Step1-6: Conduct the two scrambling processes to the target blocks.

Step1-7: Embed the payload into the target blocks in the data hiding order defined in Step 1-4.

Step1-8: Integrate all the blocks, and the output image $\tilde{I}'_{enc}$ is derived.

By switching the encryption and data hiding processes, i.e., Steps 1-6 and 1-7, we can first embed the payload into the original image and then encrypt the intermediate image. Accordingly, the data hiding domain is selectable depending on a user's request.

B. MSB-Prediction Based RDH Method

The RDH-MSB method [11] has attained high hiding capacity. The hiding capacity is around 1 bpp by introducing MSB prediction. This method adopts MSB replacement for data hiding to the available blocks that have been defined by MSB prediction. LSB replacement is more common for data hiding such as used in Ma et al.'s method [8]. However, LSBs among the neighboring pixels are typically non-correlated. So, it is difficult to attain both reversibility and high hiding capacity by using LSB replacement. In contrast, MSBs among the neighboring pixels tend to have the same value, and thus MSB prediction is more effective than LSB prediction. The outline of the RDH-MSB method is shown in Fig. 2. We describe the detailed procedure as follows.

Step2-1: Detect the MSB prediction errors in the plain domain using the neighboring pixels and store the errors in the error location binary map e. Note that the scan order of MSB prediction can be chosen from four types, that is, the raster and serpentine scan orders in the horizontal/vertical directions.

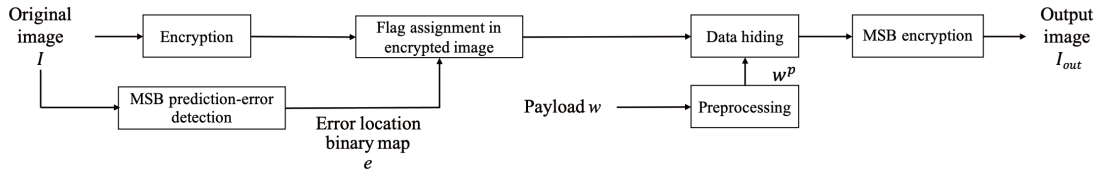


Fig. 2 Block diagram of RDH-MSB method [11].

Step2-2: Encrypt the original image I using the exclusive-or operation with the pseudo-random number sequence.

Step2-3: Divide the encrypted image and e into blocks with 8×1 pixels and 8 bits, respectively.

Step2-4: In accordance with e, if one or more prediction errors are identified in a block, exclude the block from the embeddable blocks and substitute the MSBs with the values of e. In the meanwhile, the blocks without prediction errors are defined as the embeddable ones.

Step2-5: Assign the flags to the first and final blocks of each sequence of the embeddable blocks. In particular, replace the MSBs of those blocks with 1s.

Step2-6: In each embeddable block, replace the MSBs of the eight pixels with the payload bits. Note that the payload should be preprocessed to be correctly extracted.

Step2-7: Encrypt the entire MSBs, and derive the output image I_{out}.

C. Grayscale-Based EtC System

The block scrambling based encryption method for EtC systems [14] was extended to enhance the security [16]. This method transforms an original image from RGB to YCbCr, and combines each component to derive the grayscale-based image. The three processes of block scrambling based encryption [14], i.e., position scrambling, block rotation/flip, and negative-positive transformation, are conducted to the grayscale image. The size of the encrypted image is three times as large as that of the original image, and also the smaller sized blocks can be introduced in the grayscale-based method. The key space of the encrypted image is thereby larger than that of the conventional methods [14], [15]. That contributes to enhance the robustness against brute-force attacks. Further, this method is more resistant to jigsaw solving attacks by the use of grayscale images with less color information. Hereafter, this method is called the G-EtC method. We describe each step as follows.

Step3-1: Transform the original image I from RGB to YCbCr.

Step3-2: Combine YCbCr channels into one grayscale image I'.

Step3-3: Divide I' into a certain size of blocks.

Step3-4: Encrypt each block.

Step3-5: Integrate all blocks, and the encrypted image I'_E is composed.

The block scrambling based encryption [14], [15] has another scrambling process: color component shuffling. Since the grayscale-based image is used in this method, this process is omitted. In the meanwhile, position scrambling is conducted in the larger space without the boundary of each color component. Thus, the security is not weaken without color component shuffling.

III. Proposed Method

We propose a new framework of RDH-EI, which can control both compression efficiency and hiding capacity. First, the original image is arbitrarily classified into two regions. In this paper, we assume that each target image would be divided into region of interest (ROI) and non-region of interest (non-ROI). The RDH-EtC method [12] is applied to ROI. Here, the proposed method replaces the encryption process in [12] with the G-EtC method [16]. Note that the color conversion in [16] is not performed in our method. On another front, the RDH-MSB method [11] is conducted to non-ROI. Accordingly, ROI can be compressed after encryption and also be decrypted without data extraction, while the hiding capacity is low. In contrast, non-ROI has high hiding capacity around 1 bpp, while the features in ROI would be compromised.

The RDH-MSB and RDH-EtC methods can be applied to ROI and non-ROI, respectively. Namely, these two methods can be introduced for either of both regions. We describe the detailed procedure and features of the proposed method.

A. Procedure of Proposed Method

Figure 3 illustrates the framework of the proposed method. We explain each step as follows.

Step1: Combine R, G, and B components of the original image I to obtain the grayscale-based image I_G.

Step2: Classify I_G into ROI and non-ROI, and define them as I_G^{ROI} and I_G^{nROI}, respectively.

Step3: In I_G^{ROI}, apply the RDH-EtC method [12], and $\tilde{I}_{G_{enc}}^{ROI}$ is derived.

Step4: In I_G^{nROI}, adopt the RDH-MSB method [11], and $\tilde{I}_{G_{enc}}^{nROI}$ is derived.

Step5: Intergate $\tilde{I}_{G_{enc}}^{ROI}$ and $\tilde{I}_{G_{enc}}^{nROI}$ to output the image $\tilde{I}_{G_{enc}}$.

Since ROI adopts the RDH-EtC method, we can first embed the payload and then encrypt the marked ROI. Therefore, a

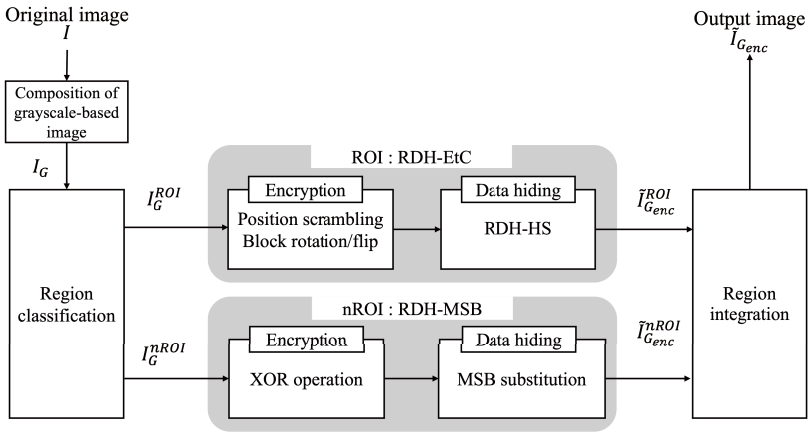


Fig. 3 Block diagram of proposed method.

user can flexibly embed the payload in plain and/or encrypted domains.

B. Contribution of Proposed Method

The proposed method can flexibly control the hiding capacity and compression efficiency by taking notice of advantages of the previous works. Here, we discuss the flaws of the previous works and prove the effectiveness of the proposed method.

1) Flaws of Previous Works: The RDH-EtC method can freely embed and extract the payload in plain and/or encrypted domains. Further, the output images are losslessly compressed by image coding standards, such as JPEG-LS [17] and JPEG 2000 [18]. This method, however, embeds the payload at a maximum hiding rate of 0.07 bpp for grayscale images. Compared to the conventional RDH-EI method [8] with a maximum rate of 0.5 bpp, the RDH-EtC method has a significantly low hiding capacity.

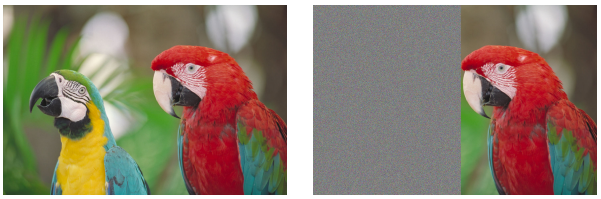
In contrast, the RDH-MSB method has quite high hiding capacity with approximately 1 bpp. The mathematical complexity is also relatively low. However, this method adopts the pixel-based encryption, and thus cannot compress the output images by using image coding standards. Additionally, the payload must be extracted in encrypted domain, and the decryption process cannot be conducted without data extraction. Although the RDH-MSB method has attained the high hiding capacity in RDH-EI, there exist multiple constraints.

The proposed method utilizes their features and controls the balance between the hiding capacity and lossless compression rate. Next, we clarify the effectiveness of our method.

2) Effectiveness of Proposed Method: The proposed method considers the flaws of the previous works and makes effective use of their advantages. The hiding capacity, which is seriously low in the RDH-EtC method, is enhanced by introducing the RDH-MSB method. As previously stated, a maximum hiding rate of the RDH-EtC method is 0.07 bpp. In contrast, the hiding capacity is approximately 1 bpp in the RDH-MSB method. When the RDH-MSB method is applied to 10% of the spacial domain in a target image, the hiding capacity for the whole of the encrypted image is improved to around 0.1 bpp without considering the capacity of the other 90% of the spacial domain. It is much larger than the maximum hiding capacity of the RDH-EtC method only. Thus, the proposed method can enhance the hiding capacity as increasing the region, where the RDH-MSB method is applied.

In the RDH-MSB method, the output image cannot be compressed by image coding standards. Our method refines the issue by introducing the RDH-EtC method. The RDH-MSB method adopts the pixel-based encryption and dose not consider the compression performance of the output images. Conversely, the output image derived by the RDH-EtC method can be losslessly compressed using JPEG-LS and JPEG 2000. In the proposed method, the region, where the RDH-EtC method is applied, can be greatly compressed, while the region, where applying the RDH-MSB method, is not effective for compression. Overall, the output image can be compressed in some degree. The compression performance would be enhanced by assigning the RDH-EtC method to wider region. Note that there is a trade-off between the hiding capacity and compression efficiency.

The proposed method can derive the marked image in ROI, where the RDH-EtC method is applied. This means that ROI can be decrypted without revealing the payload to other users. Figure 4 shows the image decrypted only in ROI; the right-side parrot has been defined as ROI. It is clear that the decrypted ROI has high quality despite containing the payload. Further, we can flexibly embed and extract the payload in ROI; thus, we suppose three kinds of models. First, an image owner embeds the payload before encryption. In another case, the third party, such as a channel provider, hides the payload, e.g., the server information and time stamps, after encryption. In the third model, by dividing ROI into two fields, the first and second models can be applied to each field, respectively. In any case, the embedded payload can be restored in plain or encrypted domain.



(a) Original image (b) Marked image

Fig. 4 Resulting image with only ROI decrypted.

IV. Experimental Results

We evaluate the effectiveness of the proposed method from the aspects of the lossless compression performance and hiding capacity. We used 24 test images from the image database [19] with $2,048 \times 3,072$ or $3,072 \times 2,048$ pixels. The original image is split into two regions horizontally; the top and bottom regions are defined as ROI and non-ROI in our simulation. To confirm the transition of the compression performance and hiding capacity, we use the variable area-ratios of ROI and non-ROI, that is, 100:0, 75:25, 50:50, 25:75, and 0:100. In practice, an user can divide the target image into these two regions more flexibly. We concatenate three color components horizontally in order from R to B. Note that arbitrary direction and order can be adopted for concatenation. The block size for encryption in ROI is 16×16 pixels. Figure 5 shows the output images obtained by the proposed method.

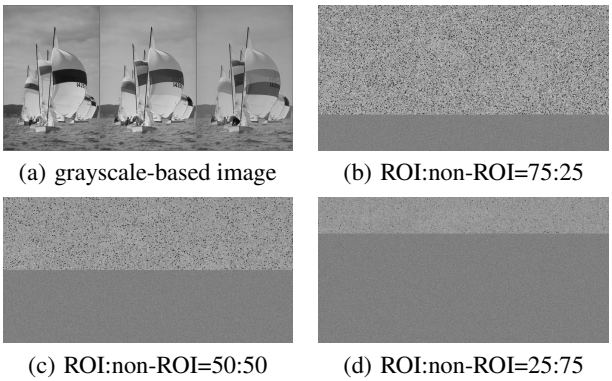
A. Lossless Compression Performance

We evaluate the lossless compression performance using JPEG-LS [17] and JPEG 2000 [18]. Figure 6 shows the bitrates of the compressed output images. It is obvious that the higher compression performance can be attained as the area of ROI becomes wide. In the case of the area ratio of 0:100, i.e., the RDH-MSB method is applied to the entire image, the bitrates by the JPEG-LS and JPEG 2000 compressions are higher than 8 bpp, respectively. This means that the data amount of the compressed image becomes higher than that of the original image. In contrast, since the RDH-EtC method is adopted to a portion of the image in the proposed method, the entire output-image can be compressed and also the compression performance is controllable.

B. Hiding Capacity

Figure 7 exhibits the hiding capacity in the proposed method. As the area of non-ROI becomes wide, our method has the higher capacity. In case that the area ratio is 100:0, that is, the RDH-EtC method is applied to the entire image, the hiding capacity is much lower than 0.1 bpp. Conversely, the RDH-MSB method attains the high capacity, which is close to 1 bpp. The proposed method can enhance the capacity up to around 1 bpp by adjusting the area ratio of ROI and non-ROI.

As can be seen from the above description, there exists a trade-off between the compression performance and hiding capacity. The most advantage of our method is that both of them can be flexibly controlled by employing the RDH-EtC and RDH-MSB methods separately for different regions. It is expected that the proposed framework would be applied to the wider fields including cloud and social networking services.



(a) grayscale-based image (b) ROI:non-ROI=75:25

(c) ROI:non-ROI=50:50 (d) ROI:non-ROI=25:75

Fig. 5 Output images by proposed method (kodim9).

V. Conclusions

We proposed a new framework of RDH-EI that considers both the compression efficiency of output images and hiding capacity. The proposed method can flexibly control the hiding capacity and compression efficiency responding to the area ratio of two regions. In one region, where the RDH-EtC method is adopted, the marked image can be derived, and the payload can be flexibly embedded and extracted in plain and/or encrypted domains. In another region, where the RDH-MSB method is applied, the hiding capacity is significantly improved. Through our experiments, we confirmed that the hiding capacity and compression performance can be controlled in proportion of the area ratio of two regions.

References

[1] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: advances in the past two decades," IEEE Access., vol.4, pp.3210–3237, 2016.

[2] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol.16, no.3, pp.354–362, 2006.

[3] M. Fujiyoshi, S. Sato, H.-L. Jin, and H. Kiya, "A Location-Map Free Reversible Data Hiding Method using Block-Based Single Parameter," in Proc. IEEE Int. Conf. Image Process., vol.3, pp.257–260, 2007.

[4] S. Weng, Y.-Q. Shi, W. Hong, and Y. Yao, "Dynamic improved pixel value ordering reversible data hiding," Inf. Sci., vol.489, pp.136–154, 2019.

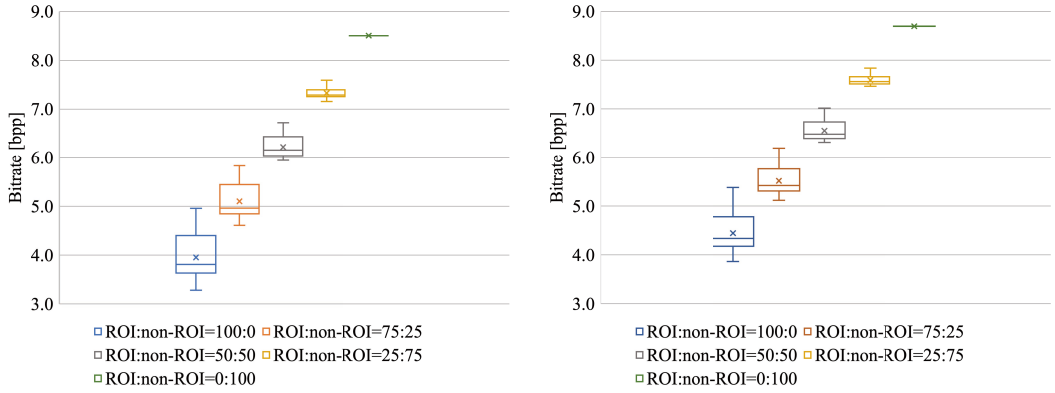
[5] J. Wang, X. Chen, J. Ni, N. Mao, and Y. Shi, "Multiple histograms-based reversible data hiding: Framework and realization," IEEE Trans. Circuits Syst. Video Technol., vol.30, pp.2313–2328, 2020.

[6] W. He, G. Xiong, and Y. Wang, "Reversible Data Hiding Based on Adaptive Multiple Histograms Modification," IEEE Trans. Inf. Forensics Security., vol.16, pp.3000–3012, 2021.

[7] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Process. Lett., vol.18, no.4, pp.255–258, 2011.

[8] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol.8, no.3, pp.553–562, 2013.

[9] P. Puteaux and W. Puech, "An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images," IEEE Trans. Inf. Forensics Security, vol.13, no.7, pp.1670–1681, 2018.



(a) JPEG-LS (b) JPEG 2000

Fig. 6 Lossless compression performance.

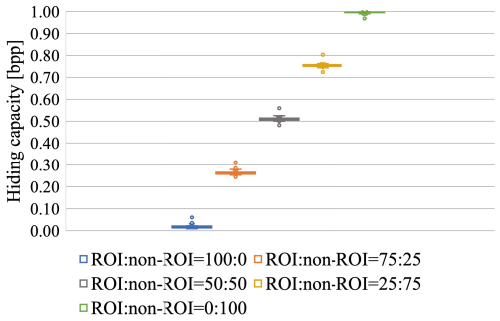


Fig. 7 Data hiding capacity.

[10] I.-C. Dragoi and D. Coltuc, "On the Security of Reversible Data Hiding in Encrypted Images by MSB Prediction," IEEE Trans. Inf. Forensics Security, vol.16, pp.187–189, 2021.
[11] R. Hirasawa, S. Imaizumi, and H. Kiya, "An MSB Prediction-Based Method with Marker Bits for Reversible Data Hiding in Encrypted Images," in Proc. of IEEE LifeTech, pp.48–50, 2021.
[12] S. Imaizumi, Y. Izawa, R. Hirasawa, and H. Kiya, "A Reversible Data Hiding Method in Compressible Encrypted Images," IEICE Trans. Fundamentals, vol.E103-A, no.12, pp.1579–1588, 2020.
[13] P. Puteaux, S.-Y. Ong, K.-S. Wong, and W. Puech, "A survey of reversible data hiding in encrypted images – The first 12 years," J. Vis. Commun. Image Represent., vol.77, pp.103085, 2021.
[14] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for JPEG/Motion JPEG standard," IEICE Trans. Fundamentals, vol.E98-A, no.11, pp.2238–2245, 2015.
[15] S. Imaizumi and H. Kiya, "A block-permutation-based encryption scheme with independent processing of RGB components," IEICE Trans. Inf. & Sys., vol.E101-D, no.12, pp.3150–3157, 2018.
[16] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-Then-Compression Systems Using Grayscale-Based Image Encryption for JPEG Images," IEEE Trans. Inf. Forensics Security, vol.14, no.6, pp.1515–1525, 2019.
[17] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. Image Process., vol.9, no.8, pp.1309–1324,2000.
[18] "Information technology – JPEG 2000 image coding system – Part 1: Core coding system," International Standard ISO/IEC IS–15444–1, Dec. 2019.
[19] [Online] https://www.math.purdue.edu/ lucier/PHOTO_CD /BMP_IMAGES/