REVERSIBLE DATA HIDING IN ENCRYPTED IMAGES USING PREDICTION-ERROR ENCODING

Shuang Yi[1,2], *Yicong Zhou*[2*]

[1]Chongqing Institution of Higher Learning Center of Forensic Science Engineering and Research, Southwest University of Political Science and Law, Chongqing 401120, China
[2]Department of Computer and Information Science, University of Macau, Macau 999078, China

ABSTRACT

In this paper, we propose a reversible data hiding method in encrypted images (RDHEI) using prediction-error encoding (PE-RDHEI). It uses a weighted checkerboard based prediction to predict 3/4 of the pixels in an original image. The obtained prediction-error values and the unmodified pixels are encrypted separately. The data hider then embed secret data into the encrypted prediction-error values using the prediction-error encoding method. At the receiver side, the secret data and original image can be completely extracted and recovered. Compared with existing RDHEI methods, PE-RDHEI significantly improves the embedding rate. Experimental results are provided to show the excellent performance of our proposed algorithm.

Index Terms— Reversible data hiding, encrypted image, prediction-error encoding, weighted checkerboard-based prediction privacy protection

1. INTRODUCTION

Reversible data hiding (RDH) [1] in digital images is a technique aims to perfectly recover the original image after the secret data has been extracted. In recent years, due to the rapid development of Cloud computing platforms and Cloud storage applications, more and more researchers show their interests to develop reversible data hiding (RDH) algorithms in encrypted images (RDHEI). In this application, both the original image and secret data need to be protected. Meanwhile, the data hider is unable to access the original image content. Thus, the original image is first encrypted and then send to the data hider for data embedding.

The early RDHEI methods [2–6] use the stream cipher to encrypt the original image, secret data is then embedded into the encrypted image by flipping several least significant bits (LSBs) in small image blocks. Data extraction and image recovery are accomplished by analyzing the smoothness of the decrypted image. Zhou *et al.* [7] use the public key modulation scheme to embed secret data into the stream-cipher-encrypted image blocks. These are joint methods that data extraction and image recovery are performed simultaneously. More applicable, some separable RDHEI methods have been developed. Method in [8] embeds secret data into the stream-cipher-encrypted image by compressing several LSB planes. Qian *et al.* [9] improved Zhang *et al.*'s method [8] by separating the LSB planes into three groups and embedding secret data separately. The original image is progressively recovered from these three groups of pixels. Xiao *et al.* [10] use the homomorphism encryption to encrypt the original image and adopt the pixel value ordering strategy for data embedding. Ma *et al.* [11] suggest to reverse a spare space from the original image before encryption. They separate the original image into two parts, the smooth area and coarse area, and embed several LSBs of the coarse area into the smooth area using the traditional RDH method, the secret data is then embedded into the reserved space. Since reserving room before image encryption is more convenient and efficient, a large embedding rate can be achieved. Mathew *et al.* [12] improved Ma *et al.*'s method by dividing the original image into small blocks and separate them into smooth and coarse areas separately. In this way, it will increase the embedding rate while reducing distortions in the recovered image. In Zhang *et al.*'s method, less than 20% of the pixels in the original image are randomly selected and predicted by their surrounding pixels and the prediction-error values are encrypted for data embedding. Cao *et al.* [13] use the sparse representation technique to compress the image and reserve room for data embedding. Papers in [14] and [15] suggest to encrypt the original image while keeping spatial correlations with small image blocks, so that the secret data can be embedded into the encrypted image blocks using the traditional RDH methods.

In this paper, we propose a RDHEI method using prediction error encoding. It uses a weighted checkerboard based prediction method to predict 3/4 of the pixels in the original image using the remaining 1/4 of the pixels. Different from the previous methods that use the traditional

This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/189/2017/A3, and by the Research Committee at University of Macau under Grants MYRG2016-00123-FST and MYRG2018-00136-FST.

Corresponding author. Tel.: +853 83978458; Fax: +853 88222426, E-mail address: yicongzhou@umac.mo

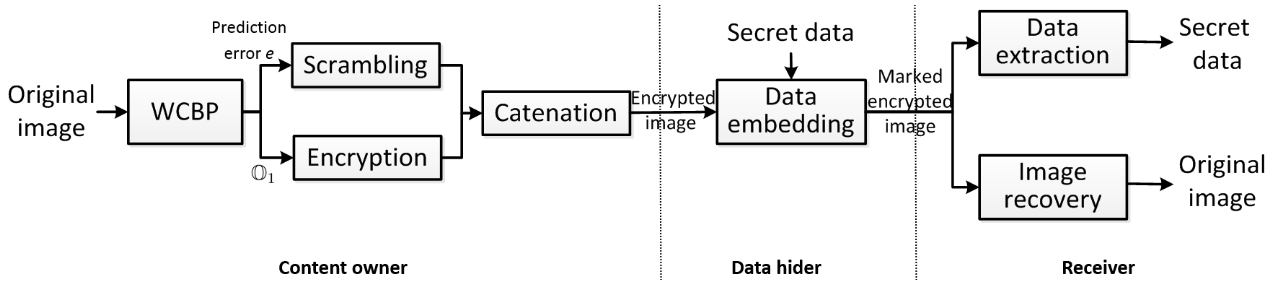

Fig. 1: Framework of PE-RDHEI.

RDH method to embed secret data into the prediction-error values by histogram shifting and expansion, we convert the prediction-errors into some specific encoded bits and embed secret data into the reserved bits by bit replacement, so that a large embedding rate can be achieved.

The rest of this paper is organized as follows: Section 2 introduces the proposed PE-RDHEI in detail. Section 3 shows the experimental results of PE-RDHEI and comparisons with several existing methods. Finally, section 4 draws a conclusion.

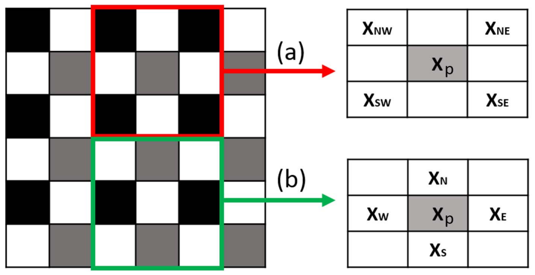

Fig. 2: Illustrative example of WCBP.

2. PE-RDHEI

The framework of the proposed PE-RDHEI is shown in Fig. 1. It consists of three phases: (1) generation of the encrypted image; (2) generation of the marked encrypted image and (3) data extraction and image recovery. These three phases are accomplished by the content-owner, data-hider and receiver, respectively.

2.1. Generation of the encrypted image

Given an 8-bit depth $M \times N$ gray-scale image $\mathbb{O}$, we first use the weighted checkerboard prediction (WCBP) to predict 3/4 of the pixels in $\mathbb{O}$ by its remaining 1/4 of the pixels. We then calculate the prediction-error values and encrypt them and the original 1/4 of the pixels in $\mathbb{O}$ separately. Finally, we catenate them to obtain the encrypted image.

2.1.1. Weighted checkerboard prediction (WCBP)

We first separate the pixels in $\mathbb{O}$ into two categories $\mathbb{O}_1$ and $\mathbb{O}_2$, where $\mathbb{O}_1$ consists of the $MN/4$ pixels that selected from every other rows and columns of $\mathbb{O}$ (the pixels marked in black in Fig. 2), $\mathbb{O}_2$ contains the remaining $3MN/4$ pixels (the pixels marked in gray and white in Fig. 2). The WCBP is to use pixels in $\mathbb{O}_1$ to predict pixels in $\mathbb{O}_2$. It consists of two steps as shown in Fig. 2: in the first step, the pixel marked in gray is predicted by its four diagonal pixels (Fig. 2(a)) using Eq. (1), where $rnd(*)$ converts the value to its nearest integer; in the second step, the pixel marked in white is predicted by its four neighbor pixels (Fig. 2(b)) using Eq. (2).

After pixel prediction, we obtain the prediction-error value e by

$$e = X - X_p \tag{3}$$

where X and X_p are the original and its corresponding predicted pixel values, respectively. Because e can either be positive or negative value, only $e \in [-127, 127]$ can be successfully stored with 8 bit by setting one bit to sign bit. For example, '0' for positive value '1' for negative value. Thus, for those e who out of the range of [-127, 127], we convert it to this range by Eq. (4) and use a binary location map L to record the positions of these error values. For example, '0' for the

$$X_p = \begin{cases} rnd(0.35 * (X_{NW} + X_{SE}) + 0.15 * (X_{NE} + X_{SW})), & \text{if } |X_{NW} - X_{SE}| < |X_{NE} - X_{SW}| \\ rnd(0.15 * (X_{NW} + X_{SE}) + 0.35 * (X_{NE} + X_{SW})), & \text{otherwise} \end{cases} \tag{1}$$

$$X_p = \begin{cases} rnd(0.35 * (X_N + X_S) + 0.15 * (X_W + X_E)), & \text{if } |X_N - X_S| < |X_W - X_E| \\ rnd(0.15 * (X_N + X_S) + 0.35 * (X_W + X_E)), & \text{otherwise} \end{cases} \tag{2}$$

Table 1: Prediction-error and its encoded results.

e	-8	-7	-6	-5	......	4	5	6	7
Encoded bits	00000	00001	00010	00011	......	01100	01101	01110	01111

original value and '1' for the converted value. Because L is important for image recovery at the receiver side, we embed it into $\mathbb{O}_1$ using the traditional RDH method such as histogram shifting and prediction-error expansion.

$$e' = \begin{cases} -e - 255, & \text{if } e_{min} \leqslant e < -127 \\ -e + 255, & \text{if } 127 < e \leqslant e_{max} \\ e, & \text{otherwise} \end{cases} \tag{4}$$

where e_{min} and e_{max} are the minimum and maximum value of e, respectively. These two values will be stored in the location map L as well for the purpose of image recovering.

2.1.2. *Image encryption*

After pixel prediction, we encrypt $\mathbb{O}_1$ and the prediction-error e' separately. Here, $\mathbb{O}_1$ can be regarded as a sub-image of $\mathbb{O}$ and with a size of $\frac{2}{M} \times \frac{2}{N}$. Note that any secure image encryption algorithm can be utilized to encrypt $\mathbb{O}_1$. For simplicity, we use the stream cipher to encrypt it using Eq. (5).

$$\hat{\mathbb{O}}_1 = \mathbb{O}_1 \oplus \mathbb{R} \tag{5}$$

where '$\oplus$' is the bit-XOR operation and $\mathbb{R}$ is an $\frac{2}{M} \times \frac{2}{N}$ pseudo random image generated by image encryption key K_e. For prediction-error e, we scramble them using K_e. Finally, we combine $\hat{\mathbb{O}}_1$ and the scrambled e to obtain the final encrypted image $\mathbb{E}$.

2.2. Generation of the marked encrypted image

After obtaining the encrypted image $\mathbb{E}$, the data hider is able to embed secret data into the scrambled prediction-error values. Firstly, the data hider separates the prediction-error values into two categories, $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$, where $\dot{\mathbf{e}}$ consists of the e that falling into the range of [-8, 7], and $\ddot{\mathbf{e}}$ contains the value of e' that in the range of $[-127, -9] \cup [8, 127]$. Only the prediction-error value in $\dot{\mathbf{e}}$ will be utilized to embed secret data. Assume that there are n values in $\dot{\mathbf{e}}$, thus, the size of $\ddot{\mathbf{e}}$ is $(3MN/4 - n)$.

To embed the secret data, firstly, for each of the e' in $\ddot{\mathbf{e}}$, we use a specific bit '1' to label it by bit replacement and keep the remaining 7 bits unmodified. The been replaced bit is preselected and stored with secret data. For each of the e' in $\dot{\mathbf{e}}$, according to the value of e', 5 bits are replaced by the specific encoding bits as shown in Table. 1, and the remaining 3 bits are reserved to embed the secret data. In order to ensure the security, the secret data is encrypted using the data hiding key K_d before embedding. Therefore, the marked encrypted image is obtained and totally $3 * n$ bits are embedded. Thus, the effective embedding rate r (bpp) is

$$r = \frac{3 * n - 3MN/4 + n}{MN}. \tag{6}$$

2.3. Data extraction and image recovery

At the receiver side, one with different keys is able to obtain different contents, the original image, secret data or both, from the marked encrypted image.

2.3.1. *Data extraction*

To extract the secret data, firstly, the receiver separates the prediction-error values into two categories, $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$, by checking their labeling bits. Then, the receiver extracts 3 bits of the payload from each prediction-error in $\dot{\mathbf{e}}$ sequentially, and extracts the secret data from payload and decrypts it using K_d to obtain the original secret data. Due to the reversibility of the data embedding procedure, the secret data can be completely extracted.

2.3.2. *Image recovery*

To recover the original image, the receiver separates the prediction-error values into two categories, $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$, as in data extraction phase. Then, using the extracted payload from $\dot{\mathbf{e}}$, the receiver recovers the been replaced 1 bit of each prediction-error value in $\ddot{\mathbf{e}}$. Next, according to the labeling bits, prediction-error values in $\dot{\mathbf{e}}$ are recovered based on the one-to-one mapping as shown in Table 1. By now, all the prediction-error values are the same as they are in the encrypted image. Then, the receiver inversely permutes the prediction-error values using K_e and decrypts the remaining pixels which formed as a sub-image $\hat{\mathbb{O}}_1$ by

$$\mathbb{O}_1 = \hat{\mathbb{O}}_1 \oplus \mathbb{R} \tag{7}$$

$$e = \begin{cases} -e' - 255, & \text{if } e' \text{ is the converted value and } e' \in [-127, -255 - e_{min}] \\ -e' + 255, & \text{if } e' \text{ is the converted value and } e' \in [255 - e_{max}, 127] \\ e', & \text{otherwise} \end{cases} \tag{8}$$

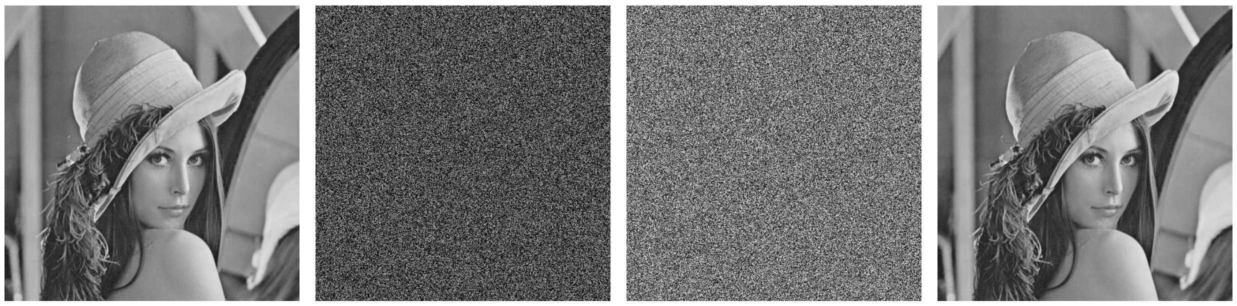
(a) (b) (c) (d)

Fig. 3: Simulation results of applying PE-RDHEI in *Lena* image: (a) the original image; (b) encrypted image; (c) marked encrypted image and with embedding rate $r = 1.907$ bpp and (d) recovered image, PSNR= $+\infty$ dB.

Table 2: Length of location map L (bits) under various images.

Images	*Lena*	*Airplane*	*Peppers*	*Barbara*	*Boat*
Length of L	0	370	0	209	0

where $\mathbb{R}$ is a pseudo randomly generated matrix using K_e. The location map L is then extracted from the sub-image $\mathbb{O}_1$ to recover the converted prediction-error to their original values by Eq. (8). Finally, the receiver calculates the prediction value X_p using sub-image $\mathbb{O}_1$ and WCBP proposed in Sec. 2.1.1, and recovers the pixel X in the prediction-error locations by

$$X = e + X_p \tag{9}$$

Hence, the original image is completely recovered.

3. EXPERIMENTAL RESULTS AND COMPARISONS

In this section, we show the experimental results of the proposed PE-RDHEI and comparisons with several related works. All test images used in this section are selected from the Miscelaneous[1] database and with a size of 512×512.

Fig. 3 shows an example of applying PE-RDHEI to the test image *Lena*. As can be seen, the encrypted image (Fig. 3(b)) and marked encrypted image (Fig. 3(c)) are both noise-like ones. The embedding rate is as large as 1.907 bpp and the recovered image is exactly the same with the original one.

Table. 2 shows the length of location map L under various images. As can be seen, due to the good performance of WCBP, the pixel in the original image can be well predicted, resulting a quit small length of L, so that it can be easily embedded into the sub-image before image encryption.

Table. 3 shows the embedding rate r (bpp) comparisons of PE-RDHEI with several related works. For those methods who cannot completely extract the secret data, the embedding rate r is reduced to $r(1 - H(\rho))$, where $H(\rho)$ is the binary entropy function with error rate ρ. For methods in [2] and [3], we set the block size to 8×8 for demonstration. For Huang *et al.*'s method, we set the block size to 2×2 and use the difference histogram shifting method for experiments. From the results we can observe that, the proposed PE-RDHEI is able to reach the maximum embedding rate among all test images, which shows the excellent performance of the proposed algorithm.

[1]http://decsai.ugr.es/cvg/dbimagenes/g512.php.

Table 3: Embedding rate r (bpp) comparisons of PE-RDHEI with several related works.

Methods	Images			
	Lena	*Airplane*	*Peppers*	*Boat*
Hong [3]	0.004	0.001	0.004	0.004
Li [5]	0.010	0.013	0.006	0.009
Zhang [8]	0.036	0.036	0.036	0.036
Huang [15]	0.103	0.147	0.077	0.061
Yin [14]	0.122	0.193	0.113	0.141
Zhou [7]	0.153	0.194	0.189	0.152
Xiao [10]	0.224	0.264	0.213	0.145
Ma [11]	0.923	1.085	0.636	0.996
PE-RDHEI	1.907	1.867	1.837	1.646

4. CONCLUSION

In this paper, we proposed an encryption domain based reversible data hiding method using prediction-error encoding. It uses a weighted checkerboard based prediction (WCBP) to predict pixel values in the original image. At the data hider side, the encrypted prediction-error values are converted into several specific 5-bit binary codes. Secret data is then embedded into the reserved 3 bit of each prediction-error value. Due to the good performance of WCBP and prediction-error encoding technique, the proposed PE-RDHEI is able to reach a large embedding rate. Experimental results and comparisons have shown the excellent performance of our proposed algorithm.

5. REFERENCES

[1] Zhicheng Ni, Yun-Qing Shi, N. Ansari, and Wei Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[2] Xinpeng Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[3] Wien Hong, Tung-Shou Chen, and Han-Yan Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[4] Jie Yu, Guopu Zhu, Xiaolong Li, and Jianquan Yang, "An improved algorithm for reversible data hiding in encrypted image," *Digital Forensics and Watermaking*, vol. 7809, pp. 384–394, 2013.

[5] Ming Li, Di Xiao, Zhongxian Peng, and Hai Nan, "A modified reversible data hiding in encrypted images using random diffusion and accurate prediction," *ETRI Journal*, vol. 36, no. 2, pp. 325–328, 2014.

[6] Xin Liao and Changwen Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, no. 0, pp. 21–27, 2015.

[7] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[8] Xinpeng Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[9] Z. Qian, X. Zhang, and G. Feng, "Reversible data hiding in encrypted images based on progressive recovery," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1672–1676, 2016.

[10] Di Xiao, Yanping Xiang, Hongying Zheng, and Yong Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *Journal of Visual Communication and Image Representation*, vol. 45, no. Supplement C, pp. 1–10, 2017.

[11] K. 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, pp. 553–562, 2013.

[12] T. Mathew and M. Wilscy, "Reversible data hiding in encrypted images by active block exchange and room reservation," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 839–844.

[13] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[14] Zhaoxia Yin, Huabin Wang, Haifeng Zhao, Bin Luo, and Xinpeng Zhang, "Complete separable reversible data hiding in encrypted image," *Cloud Computing and Security: First International Conference, ICCCS 2015*, pp. 101–110, 2015.

[15] F. Huang, J. Huang, and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.