Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8847559, 10 pages
https://doi.org/10.1155/2020/8847559

Research Article

Reversible Information Hiding Algorithm Based on Multikey Encryption

Zhaohui Li,[1,2] Yiqing Wang,[2] Zhi Wang,[1,2] Zheli Liu,[1,2] Jian Zhang,[1,2] and Min Li[1,2]

[1] *Tianjin Key Laboratory of Network and Data Security Technology, College of Cyber Science, Nankai University, 300350 Tianjin, China*
[2] *College of Computer Science, Nankai University, 300350 Tianjin, China*

Correspondence should be addressed to Min Li; limintj@nankai.edu.cn

Received 25 June 2020; Revised 30 August 2020; Accepted 20 October 2020; Published 16 November 2020

Academic Editor: Ashok Kumar Das

This paper proposes a scheme of reversible data hiding in encrypted images based on multikey encryption. There are only two parties that are involved in this framework, including the content owner and the recipient. The content owner encrypts the original image with a key set which is composed by a selection method according to the additional message. Thus, the image can be encrypted and embedded at the same time. Additional message can be extracted given that the recipient side could perform decryption strategy by exploiting spatial correlation; then, original image can be recovered without any loss. Compare with other current information hiding mechanism, the proposed approach provides higher embedding capacity and is also able to perfectly reconstruct the original image as well as the embedded message. Rate distortion of the proposed method outperforms the previously published ones.

1. Introduction

Nowadays, information hiding is gaining considerable attention. In the cloud computing environment, in order to ensure the information security, the uploaded images generally need to be encrypted in advance. Some images also need to embed some additional information by information hiding technology. For example, medical images often need to embed patient's name, doctor's name, medical records, and other information. We can all acknowledge that information security is absolutely necessary in our daily life, and information hiding algorithm brings a new solution to this issue. The traditional scheme is to embed information into the original image and then encrypt the image. The scheme based on multikey encryption proposed in this paper can realize embedding and encrypting at the same time that is to encrypt the image and embed information at the same time.

Different from watermarking, correctly recover the original image comes to the priority [1]. It has been widely needed in traditional field such as military, medical, and legal situations where people pursue extremely consistency between the original image and decrypted image. Meanwhile, it is getting more widespread in the cloud computing domain which is representing the growing tendency of the computing industry [2, 3]. Various privacy-preserving applications and cloud computing greatly stimulate people's research interest on signal processing over encrypted domain [4–6]. In most cases, such as cloud computing and secure remote sensing, participants who process the image cannot be fully trusted. So, there are concerns about public security and privacy, and it is safer for the image to be encrypted before sending forward. For example, in a cloud computing case, if the content owner who wish to send information to remote server without disclosure, he or she can embed the secret information in an image to make the original image encrypted. For the recipients, users could download the encrypted image and then use certain solutions to extract the information.

The purpose of reversible data hiding is to recover the embedded bits while the original image can be correctly reconstructed. When the data hiding is performed with a reversible technique, the recipient can perfectly restore the original cover content after data extraction. Majority of former information hiding mechanisms embed the additional bits into original images directly, in other words, the embedding is performed over the plaintext domain. Existing reversible data hiding methods can be divided into three categories: histogram modification methods, lossless compression-based method, and the difference expansion methods [7]. The lossless compression-based methods make full use of the compression algorithm to vacate certain space for information embedding which was widely used in early information hiding research. However, it turned out that the capacity of embedding performance of this kind is still needs to be improved. As the second category, histogram modification methods can perform better embedding rate by shifting data form peak to its bottom in the histogram. The lately difference expansion methods generate a least significant bit (LSB) layer through doubling the difference between two adjacent pixels in order to free space for embedding.

In this paper, we bring an algorithm based on multikey encryption forward, which allows the content owner encrypt the original image while embedding the additional information. Our proposal uses a key selection mechanism to embed additional information, and data extraction can be perfectly done by exploiting the statistical spatial feature. The data embedding process is done via additional information to locate random sequence which is the same size as original image block by block. As for the recipients, with the help of spatial correlation in each block, original image can be perfectly reconstructed, and the additional information can be extracted without any loss. Numerous experimental results indicate that our proposal possess superiority among the state-of-the-art methods.

The rest of the article is structured as follows. Section 2 briefly overviews the related work on information hiding over the encrypted domain. Section 3 will elaborate the proposed system. Section 4 presents the experimental result. Finally, this paper is concluded in Section 5.

2. Related Work

Plentiful experiments have been made in the reversible data hiding area. In Zhang's proposal [8], content owner encrypts the image and submits it to the server, leaving the embedding part to the data-hider side. The data hider flips three least significant bit (LSB) layers of half of block to embed the additional information into encrypted image. Then, the recipients could extract the secret information as well as recover the original image from the processed image by using both extraction key and decryption key. The embedding rate in this method was not relatively outstanding; besides, distortion rate of recovery image is also rising while the side length of each block descends. In [9], Hong et al. enhanced Zhang's method through a side-match algorithm and spatial correlation between two adjacent blocks in order to improve the embedding rate.

However, error rate is much affected in the high activity area due to the local smoothness, which leads to [10]; this method was moving forward by Qian and Zhang; their innovation is that the data hider divided the encrypted image into three different sets and embedded the additional information into each set which means it also need three rounds to decrypt the image. We can draw a conclusion that the ultimate error rate was much affected by the first and second round of decryption and also fluctuated with the variation of segment size.

In [11], Ma et al.'s proposal was to vacate space to embed information before encryption, which confirmed to be an efficient solution for shifting fractional embedding task into encryption part. Lately, Xiong et al. in [12] designed a method to embed additional information by processing the image through integer wavelet transform (IWT) firstly. Results show that it did improve the decryption correction rate which can be observed from the chart. However, the number of embedding bits is limited.

Other related mechanisms were presented in [13–15]. Among them, [13, 14] are mainly focus on medical images which is widely needed since people normally reluctant to let other people know their illness. In [13], the reversibility was accomplished by transforming a pixel in original image into a 2×2 block, and it can surely achieve relatively high embedding rate. In [14], region of interest (ROI) was brought into their methods, and the preprocessing and contrast enhancement were performed only in ROI. So, shifting of histograms of the background pixels is not involved.

[16] is also using an algorithm that combines the image encryption and information into one single step; however, with different decryption strategy, our method brings higher performance. Distributed source coding was involved in [17]; they compress a series of bits which were selected by low-density parity check codes to make room for the secret data.

[18] is based on Tromino scrambling and adaptive pixel value ordering; they divided the image into three pixels in L shape and then encrypted the image in sequence, and they embed the information through an adaptive pixel value ordering (PVO) scheme. [19] is mainly focus on 2D vector graphics by using a key to scramble the polar angles of the vertices to encrypt the graphics. It has good performance dealing with normal operations such as rotation, scaling, translation (RST), and entity reordering.

To summarize what we discussed above, in all the reversible data hiding schemes, the private extra data hiding key is necessarily involved to make sure that embedding is performed security. We cannot help wondering, is there any solution to combine the encryption part and embedding part into only one operation, to avoid the cost of transferring two kind of key: encryption key and embedding key, while still maintain the security of system. To take all these concerns into consideration, we propose this algorithm based on multikey encryption which allows the content owners embed the information and encrypt original image at the same time by using only one key group, greatly enhanced the embedding rate. Also, the error rate in the whole procedure is minimized.

3. Proposed System

Sketch of the proposed program is given in Figure 1, which is consists of two parts: (1) data embedding and image encryption; (2) data recovery and image decryption.

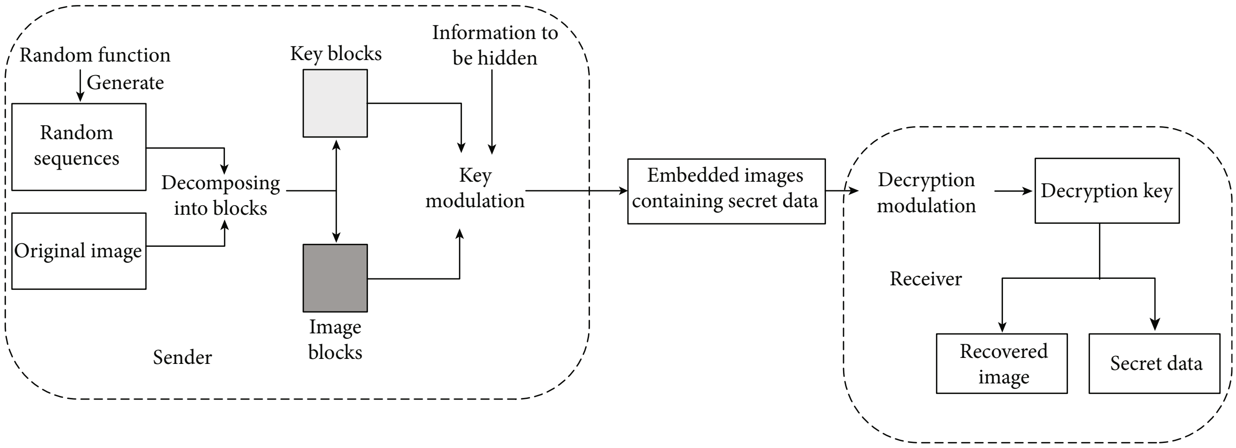


Figure 1: Sketch of the proposed method.

In phase I, the content owner encrypts original image using a stream cipher selected from a huge key group by the additional information that is how we manage to make encryption and embedding into one move. It greatly simplifies the encryption procedure and highly improves the system security.

In phase II, when the recipients obtain both encrypted image and the key group, they can make full use of spatial correlation to recover the image and to extract the additional information. As we can see, unlike majority of existing reversible data hiding scheme, the data hider is unnecessary in our proposal.

3.1. Data Embedding and Image Encryption. Similar to other reversible data hiding schemes, the stream cipher in the standard format is still used in this algorithm, such as the RC4 and AES in the CTR mode (AES-CTR). With the stream cipher, encrypted image is generated by

$$J = \mathrm{Enc}\,(X, K) = X \oplus K, \tag{1}$$

where J and X represent the encrypted and original image; in this case, K denotes the key generated by random function. Naturally, the original image can be reconstructed by

$$X = \mathrm{Enc}\,(J, K) = J \oplus K. \tag{2}$$

Assuming the original image is in uncompressed format and each pixel with gray value falling into [0,255]. When the content owner enciphers an image X sized $M \times N$, the first thing is divided the original image X into several blocks; each of these blocks is sized $m \times m$ and not overlapping spatially. Then, all these blocks are classified into two sets, as illustrated in Figure 2; blocks in the horizontal uppermost row and the vertical leftmost column are divided into the gray set, and remaining blocks are divided into the white set. For the blocks in gray set, special purpose is included which will be thoroughly discussed later.

For each block in the white set, the steps for performing the message embedding are summarized as follows:

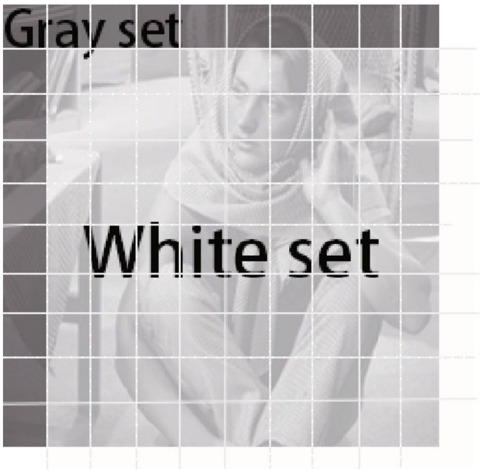


Figure 2: Example of image decomposing. For example, an image is sized 100 × 100 pixels and is divided into 10 × 10 sized block; blocks colored in gray are saved for special use, and blocks colored in white are used to embed data.

Step 1. Fetch b bits from the additional information by terms, convert this bit string into decimal, denoted by p_i.$(0 < p_i < 2^b - 1)$.

Step 2. Find the key$_i$ according to p_i, for example, when $n = 5$ and first five bits in the additional information is 00010, which makes $p_i = 2$, then the corresponding key is key$_2$.

Step 3. Embed the additional information by XOR operation between original image block and the same position in key$_i$, by now, both encryption and embedding are simultaneously finished.

For the blocks in white set, fixed b bits are embedded into each block, to performed the encryption and embedding efficient, we generate $S = 2^b$ random sequence key$_1$, key$_2$, …, key$_s$, which is also at the length of $M \times N$ bits. In this way, every successive b bit information can convert into its decimal. According to this decimal number, a special key corresponding to this number will be located. Apparently,

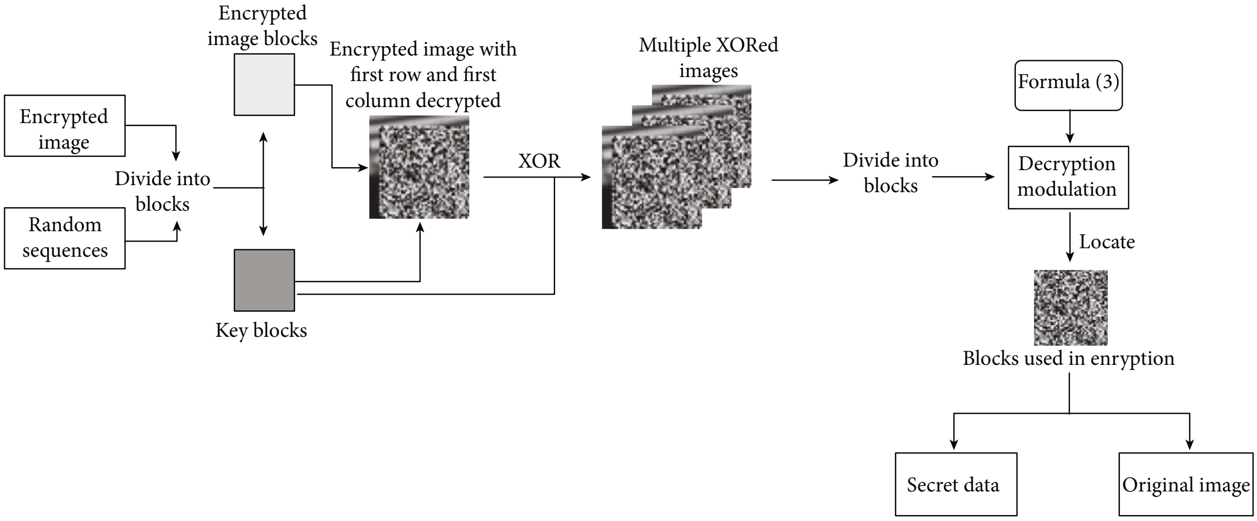


Figure 3: Comprehensive decryption steps.

each block in white set is encrypted by different random sequence, and even in some rare occasion, the decimal number is the same in two blocks; the position is still different. Given the number of blocks in the white set is W, the capacity of embedding in our method can be described as $b \cdot W$ bits.

Blocks in gray set are encrypted in the same way as the white set; the only difference is that the encryption keys are chosen by the content owner, that is, the content owner can pick his or her favor noted p_0 between 1 and 2^b, any integer he or she wants, and then employ the key key_0 correspond with p_0 into the XOR operation. Explicit procedures are presented in this page.

3.2. Data Extraction and Image Recovery. On receiver side, the hidden data can be extracted, and original image can be fully recovered using the key group we mentioned previously. To this end, we need to identify which key was used to encrypt the original image. One procedure must be finished before data extraction that is the calculation of spatial correlation. By measuring the absolute difference between two adjacent pixels, we can depict the smoothness of an image in a small range.

The concrete steps of image decryption and information extraction are as follows:

Step 1. Block the encrypted image in the same way as the encryption section does.

Step 2. Employ the selected key to encrypt blocks in gray set and generate mask_0 which first row and column are decrypted perfectly, and the remain part is still encrypted.

Step 3. Use the encrypted image performing XOR operation with key_1, key_2, ..., key_s, respectively, and then keep these S images in array XORed which sized $S \times M \times N$.

Step 4. Replace images in XORed for first column and first row with mask_0, that is, in array XORed, there are S images, and each of is partially decrypted.

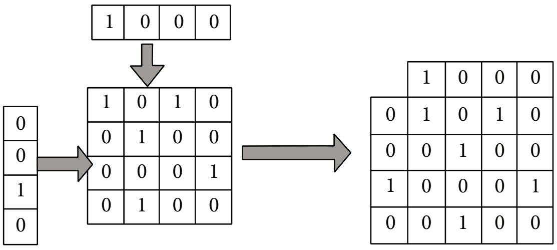
Step 5. For each block in white set, use equation (3) to calculate the absolute difference in each block. However, as it can be illustrated in Figure 3, we put an extra row and column in the calculation to enhance the matching accuracy.

Step 6. For each block in white set, the smallest absolute difference value is been looking for and written in sequence by order; the decryption mask is produced.

Step 7. With the sequence in Step 6, the additional information can be recovered losslessly, and the original image is also reconstructed by XOR operation between the decryption mask and encrypted image. Also, reconstructed by XOR operation between the decryption mask and encrypted image.

In general, more complex images tend to enlarge the summation of absolute differences. For instance, in a natural image, the pixel value differs from each other and that means the absolute difference in a natural image is relatively higher. An excellent data hiding algorithm can disturb most of pixel values among the images very well which make the summation of absolute differences particularly small. In this case, the encrypted blocks, compared with unencrypted blocks, are inclined to reach a more uniform distribution. Through this feature, we can clearly distinguish which key was used in encryption. And then, according to the key number, we can perfectly retrieve the additional information.

One thing we must take into account is that, with the block size getting smaller, the embedding capacity increases; however, it also indicates that the number of available samples in a block declines, which potentially causes the deviation of calculation. Zhou et al. in [16] also proposed a method to embed information by cutting the whole image into blocks, but in decryption side, Zhou et al.'s choice was



(a) (b)

Figure 4: Example of blocking, suppose we intend to set the block sized 4 × 4 pixels, however, the calculation of absolute difference is performed in a bigger block.

to pick four neighbors around the central pixel as in northeast, southeast, south, and east.

In [9], Hong and their team improved Zhou et al.'s proposal by calculating the summation of the horizontal absolute differences and vertical absolute differences of pixels in one block using the following formula:

$$f = \sum_{x=1}^{n} \sum_{y=1}^{n-1} \left| p(x,y) - p(x,y+1) \right| + \sum_{x=1}^{n-1} \sum_{y=1}^{n} \left| p(x,y) - p(x+1,y) \right|. \quad (3)$$

Among them, $p(x,y)$ represents the pixel value at coordinates (x,y), and f represents the sum of the pixel differences between any two adjacent pixels of the image block.

Lossless recovery of the original image and perfectly reconstruct the information would be the ultimate pursuit for every reversible data hiding method. Earlier, we talked about blocks in gray set are saved for special purpose; these blocks can be decrypted absolutely correct for the index of keys is chosen by the content owner and then transmitted to the recipient.

Here, we make full use of this feature through taking the extra one row and one column into consideration in each block as depicted in Figure 4. For example, we intend to set the block size 4 × 4; in Hong et al.'s proposal, the calculation of absolute differences is restricted in this 4 × 4 block as of Figure 4(a) but of Figure 4(b), with an extra column and row; the calculation is applied in a nearly 5 × 5 size block, with more pixels in one block comes more convincible results. That is why we save the first row and first column for a selected key in image encryption. In this way, the first row and first column in encrypted image can be guaranteed that the decryption is completely correct. The comprehensive decryption steps are depicted in Figure 3.

4. Experimental Result

In this section, we conducted ample experiments to evaluate the reversible data hiding method we proposed. Four grayscale images of size 512 × 512 are being experimented, including Goldhill, Baboon, Babara, and Lena as the test image. Experimental results on Lena are shown in Figure 5. We embedded 102010 bits of additional information into the original image by dividing the image with 5 × 5 block and then hiding 10 bits into each block. In Figure 5, (a) shows the original image of Lena, and (b) shows Lena after encryption; (c) is decrypted version of Lena. (d–f) show three version of Goldhill; (d) is the original one; (e) is what Goldhill looks like after encryption; (f) is after decryption. (g–i) and (j–l) represent three version of Barbara and Baboon in the same order.

From Figure 6, we can recognize that the encrypted images are well-distributed than the original one, which greatly enhanced the security level of our algorithm. This method creates proper confusion at image space and makes the number of pixels in each gray value balanced so that the attacker cannot obtain any useful information by just analysing the pixel distribution.

We conducted series of experiments to prove that no matter which carrier we use, the decryption accuracy tends to fall down along with increasing embedding rate, as it shown in Figure 7. However, exactly how much of accuracy we lose is different from pictures to pictures. In our experiment, picture Baboon has the most sophisticated geometric configuration, so it shows in Figure 8. Figure 8 depicts the relationship between embedding rate and error rate; it is obvious that image Lena has more plain geometric configuration and lower error rate under multiembedding rate situation.

As we have mentioned previously, the standardized encryption method is still used in our experiments; to fairly demonstrate the capability, the proposed scheme is compared with other data hiding algorithms that also employ the standardized encryption methods. In Table 1, we evaluate the embedding rate and data extraction error rate of other methods and our method under different block size. As it can be revealed that in all these methods, along with embedding rate increase, the error rate goes up. In contrast, our proposal provides a much higher accuracy for all block sizes.

In fact, the distortion of decryption image can only be discovered when the embedding rate is reach to 1.0981. It can be observed that, in 4 × 4 and 5 × 5 situation, the decryption can be completely reversible, but in [8, 9], the error rate is shockingly reach to 26.0346. For [16], the error rate is particularly small but at the cost of low embedding rate. Statistically, the proposed method still outperforms previous ones.

Furthermore, in Figure 9, we give the comparisons on Lena and Baboon among different algorithms for other

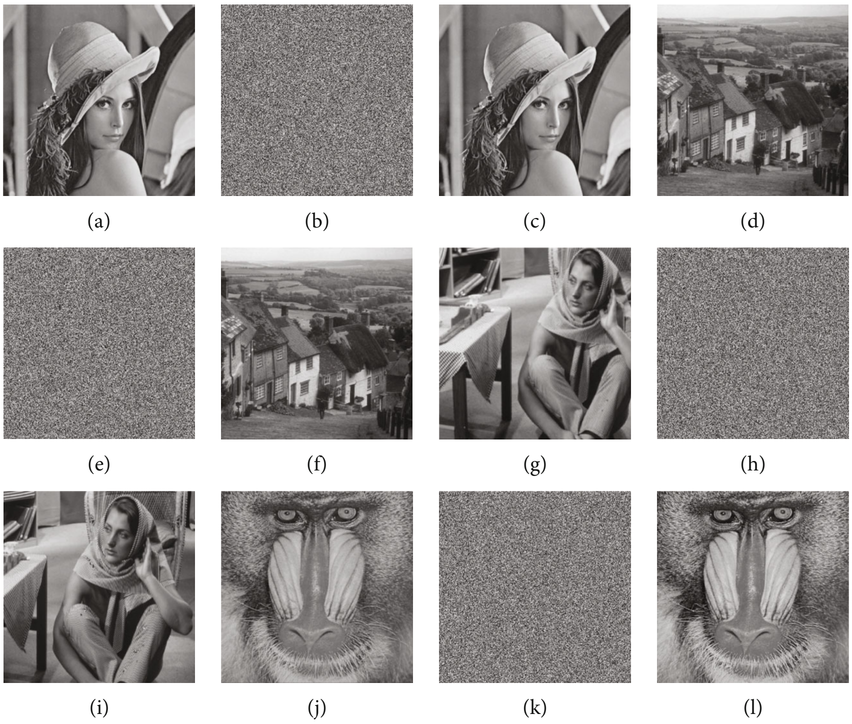
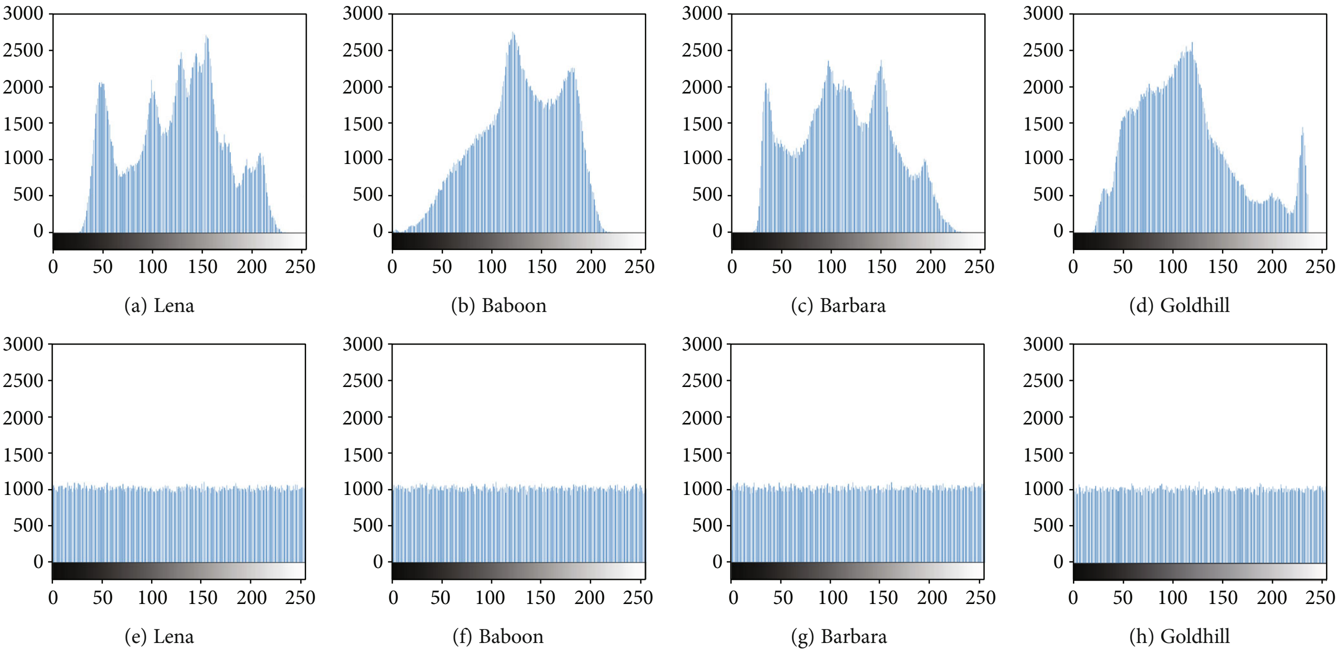

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

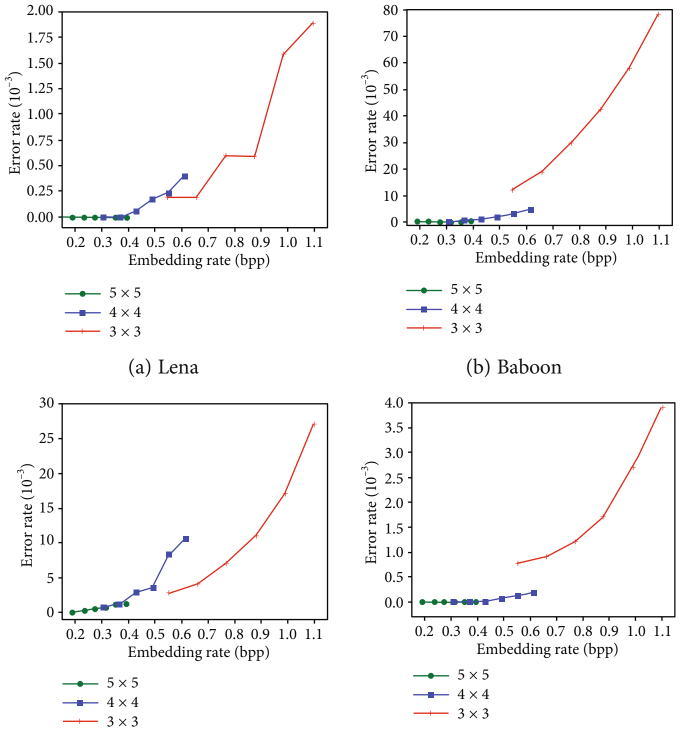
Figure 5: Experimental result on Lena and Goldhill when the block sized 5 × 5 and 10 bits of additional information were embedded in each block. (a) is the original image of Lena; (b) is the encrypted image of Lena; (c) is the decrypted image of Lena. (d) is the original image of Goldhill; (e) is the encrypted image of Goldhill; (f) is the decrypted image of Goldhill. (g) is the original image of Barbara; (h) is the encrypted image of Barbara; (i) is the decrypted image of Barbara. (j) is the original image of Baboon; (k) is the encrypted image of Baboon; (l) is the decrypted image of Baboon.



(a) Lena (b) Baboon (c) Barbara (d) Goldhill
(e) Lena (f) Baboon (g) Barbara (h) Goldhill

Figure 6: Histogram comparison between the original image and encrypted image. (a, e) are histogram comparison of Lena. (b, f) are histogram comparison of Baboon. (c, g) are histogram comparison of Barbara. (d, h) are histogram comparison of Goldhill.

approaches. Peak Signal-to-Noise Ratio (PSNR) is the ratio between maximum possible power of the signal and the destructive noise power that affects its representation accuracy, which is used to measure the result of image restoration extensively. The higher PSNR value means the better decryption.



(a) Lena

(b) Baboon

(c) Barbara

(d) Goldhill

Figure 7: Error rate under various embedding rate in (a) Lena, (b) Baboon, (c) Barbara, and (d) Goldhill.

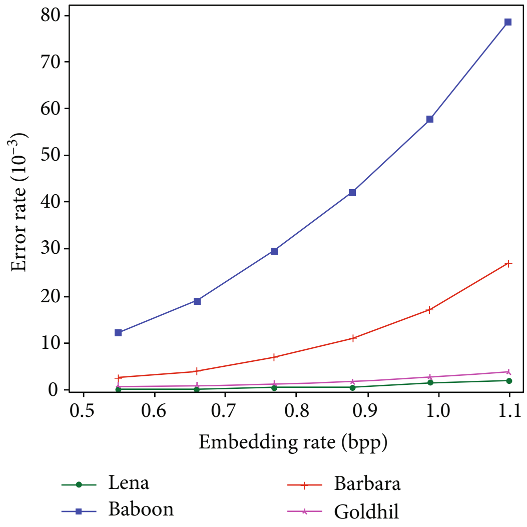


Figure 8: Error rate—embedding rate curve with different carrier.

For the record, in Figure 9(a), because the recovery version is identical to the original image, PSNR is infinite when bpp reaches to 0.392 on Lena. As shown in Figure 9, the embedding rate we perform is outstanding. Given the embedding rate range methods in [11, 15, 17] can achieve, the growth of PSNR is considerably high. On the other hand, while we maintain this superior embedding rate, the PSNR is still excellent than others.

In Figure 10, we give PSNR comparison between four different images; as we can see, PSNR value varies in different images on smoothness, the more smooth image tends to achieve higher PSNR after decryption. In Lena and Goldhill, PSNR reaches to infinite when 102010 bits (0.392 bpp) were embedded while the lowest PSNR value appears on Baboon when the embedding rate is 1.0985. But in general, the PSNR value tends to decrease along with ascending of embedding rate.

Compared to PSNR, SSIM (Structural Similarity) is more close to human perception for it takes luminance, contrast, and structure into account. The computational formula of SSIM is as below, equation (4) is how we get our SSIM value and equation (5) to equation (7) is how we measure luminance, contrast, and structure.

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma}, \tag{4}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \tag{5}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \tag{6}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}. \tag{7}$$

The value of SSIM is usually a float number between 0 and 1; the higher SSIM indicates higher decryption quality. We can say that this data hiding algorithm is reversible if the SSIM reach to 1.

Table 1: Embedding capacity (bpp) and error rate comparison.

Block size	Proposed		[8]		[9]		[16]	
	Capacity	Error rate	Capacity	Error rate	Capacity	Error rate	Capacity	Error rate
3×3	1.0981	0.0278	0.1102	35.9868	0.1102	23.1781	0.3307	0.1776
4×4	0.6153	0.0036	0.0625	26.0346	0.0625	17.6103	0.1875	0.0239
5×5	0.3922	0.0005	0.0498	22.8978	0.0498	15.6773	0.1494	0.0097

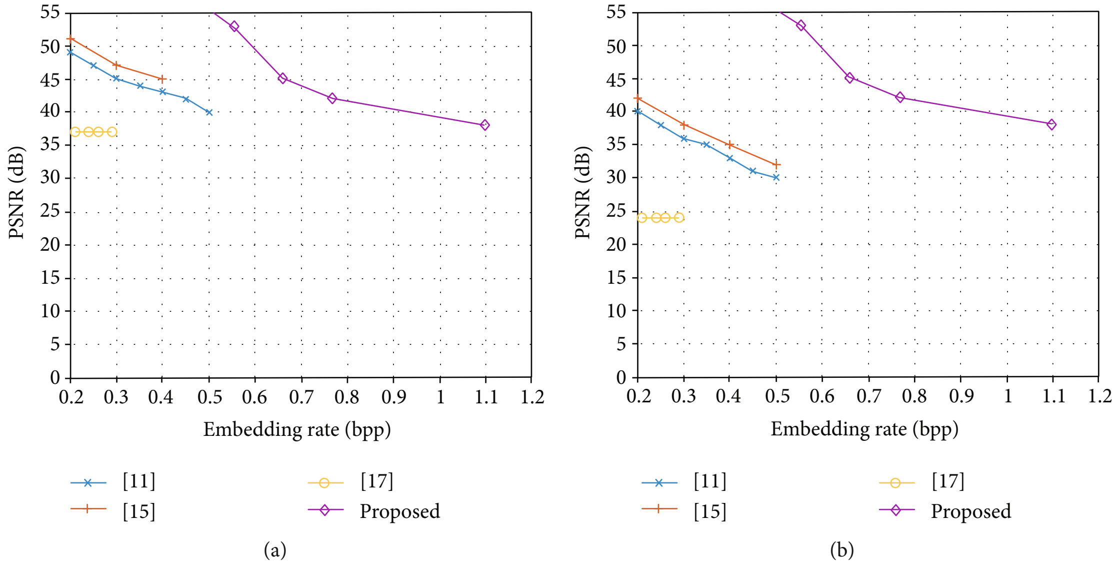


Figure 9: PSNR comparison with methods [11, 15, 17] on Lena and Baboon: (a) Lena; (b) Baboon.

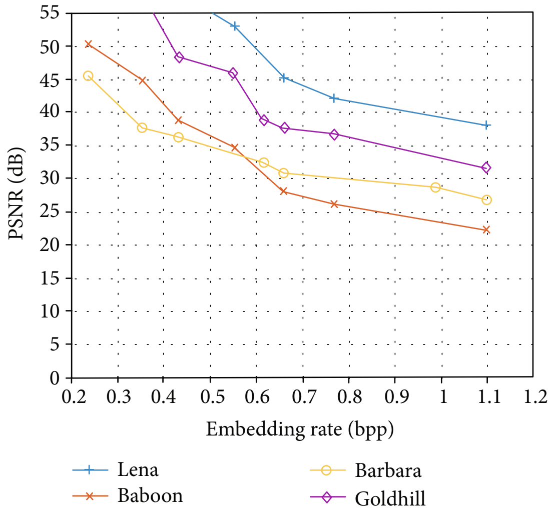


Figure 10: Comparison of rate-distortion performance indifferent images.

Tables 2–5 list the SSIM value of our method with various carrier and under various block size; easily recognized that with image Lena and image Goldhill, the number of 1.0000 value of SSIM is much more than image Baboon and image Barbara. In general, the SSIM of our method is no less than 0.88 which present a superior performance.

Table 2: SSIM table of Lena.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4×4	1.0000	0.9999	1.0000	0.9999	0.9997	0.9992
3×3	0.9987	0.9985	0.9968	0.9978	0.9938	0.9942

*5 × 5, 4 × 4, and 3 × 3 are the size of each block. **5 bits, 6 bits,..., 10 bits are how much of secret information embedded in each block.

Table 3: SSIM table of Goldhill.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4×4	1.0000	1.0000	0.9999	1.0000	1.0000	0.9990
3×3	0.9946	0.9934	0.9909	0.9897	0.9845	0.9806

*5 × 5, 4 × 4, and 3 × 3 are the size of each block. **5 bits, 6 bits,..., 10 bits are how much of secret information embedded in each block.

5. Conclusion

In this paper, we designed a novel reversible data hiding method based on multikey encryption. Instead of other reversible data hiding methods using two kinds of keys in the encryption and embedding part (embedding keys and encryption keys), we make this procedure simpler.

Table 4: SSIM table of Baboon.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5 × 5	1.0000	1.0000	1.0000	1.0000	0.9996	0.9996
4 × 4	0.9994	0.9993	0.9986	0.9971	0.9945	0.9905
3 × 3	0.9819	0.9721	0.9573	0.9403	0.9193	0.8950

*5 × 5, 4 × 4, and 3 × 3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

Table 5: SSIM table of Barbara.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5 × 5	0.9999	0.9999	0.9993	0.9993	0.9982	0.9978
4 × 4	0.9987	0.9973	0.9960	0.9959	0.9900	0.9860
3 × 3	0.9953	0.9936	0.9894	0.9840	0.9700	0.9636

*5 × 5, 4 × 4, and 3 × 3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

Here, we stick to the standard format of stream cipher; with the key modulation mechanism, the encryption and embedding can be accomplished simultaneously by simply one XOR operation. For the recipient, the original image and additional information can be obtained with key group.

We think the advantage of multikey system is that, firstly, the original image is divided into blocks, and then, different keys are selected for encryption according to the information to be embedded. The greater of the number of keys, the greater amount of information embedded in each block. That is the advantage of the algorithm using multiple keys. Also, without participation of data hider, the system becomes more secure and efficient. Both encryption and embedding can be performed at the same time at the content owner side which is not involving the third party. The additional information and original image can be safely restricted to content owner and the recipient.

We have also conducted a series of experiments to verify the outstanding performance of the proposed system. To compare with other blocking mechanism in reversible data hiding domain, our proposal not only achieved higher embedding rate but also immensely improved the decryption accuracy.

Data Availability

No data were used to support this study.

Conflicts of Interest

No conflict.

Acknowledgments

This work was supported by the 2019 Tianjin New Generation AI Technology Key Project (No. 19ZXZNGX00090), the National Natural Science Foundation of China (No. 61672300, No. 61872202), and the National Natural Science Foundation of Tianjin (No. 18ZXZNGX00140).

References

[1] T. Kalker and F. M. J. Willems, “Capacity bounds and constructions for reversible data-hiding,” in *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, pp. 71–76, Santorini, Greece, 2002.

[2] H. Wang, W. Zhang, and N. Yu, “Protecting patient confidential information based on ECG reversible data hiding,” *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13733–13747, 2015.

[3] F. Zhangjie, X. Sun, Q. Liu, L. Zhou, and J. Shu, “Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing,” *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[4] M. Barni, P. Failla, R. Lazzeretti, A. Sadeghi, and T. Schneider, “Privacy-preserving ECG classification with branching programs and neural networks,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 452–468, 2011.

[5] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk, “Generating private recommendations efficiently using homomorphic encryption and data packing,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1053–1066, 2012.

[6] T. Bianchi, A. Piva, and M. Barni, “Composite signal representation for fast and storage-efficient processing of encrypted signals,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 180–187, 2010.

[7] X. Zhang, Z. Qian, G. Feng, and Y. Ren, “Efficient reversible data hiding in encrypted images,” *Journal of Visual Communication and Image Representation*, vol. 25, pp. 322–328, 2014.

[8] X. Zhang, “Reversible data hiding in encrypted images,” *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[9] W. Hong, T. Chen, and H. Wu, “An improved reversible data hiding in encrypted images using side match,” *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[10] 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.

[11] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. 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] L. Xiong, Z. Xu, and Y. Shi, “An integer wavelet transform based scheme for reversible data hiding in encrypted images,” *Multidimensional Systems and Signal Processing*, vol. 29, pp. 1191–1202, 2018.

[13] S. A. Parah, F. Ahad, J. A. Sheikh, and G. M. Bhat, “Hiding clinical information in medical images: a new high capacity and reversible data hiding technique,” *Journal of Biomedical Informatics*, vol. 66, pp. 214–230, 2017.

[14] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, “Rate and distortion optimization for reversible data hiding using multiple histogram shifting,” *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315–326, 2017.

[15] G. Gao, X. Wan, S. Yao, Z. Cui, C. Zhou, and X. Sun, “Reversible data hiding with contrast enhancement and tamper localization for medical images,” *Information Sciences*, vol. 385, pp. 250–265, 2017.

[16] 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.

[17] Z. Qian and X. Zhang, “Reversible data hiding in encrypted images with distributed source encoding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[18] M. Long, Z. Yu, X. Zhang, and F. Peng, “A separable reversible data hiding scheme for encrypted images based on Tromino scrambling and adaptive pixel value ordering,” *Signal Processing*, vol. 176, article 107703, 2020.

[19] F. Peng, W.-y. Jiang, Y. Qi, Z.-x. Lin, and M. Long, “Separable robust reversible watermarking in encrypted 2D vector graphics,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2391–2405, 2020.