Hindawi
Complexity
Volume 2020, Article ID 6989452, 12 pages
https://doi.org/10.1155/2020/6989452

Research Article

High-Capacity Reversible Data Hiding in Encrypted Images by Information Preprocessing

Xi-Yan Li,[1] **Xia-Bing Zhou**,[2] **Qing-Lei Zhou**,[1,3] **Shi-Jing Han**,[1,4] **and Zheng Liu**[1,5]

[1] *State Key Laboratory of Mathematical Engineering and Advanced Computing, Information Engineering University, Zhengzhou 450001, China*
[2] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[3] *School of Information Engineering of, Zhengzhou University, Zhengzhou 450001, China*
[4] *Nanning Normal University, Nanning 530000, China*
[5] *Henan University of Animal Husbandry and Economy, School of Information on Engineering, Zhengzhou 450001, China*

Correspondence should be addressed to Zheng Liu; lz@hnuahe.edu.cn

Received 30 June 2020; Revised 11 September 2020; Accepted 5 October 2020; Published 22 October 2020

Academic Editor: Toshikazu Kuniya

With the development of cloud computing, high-capacity reversible data hiding in an encrypted image (RDHEI) has attracted increasing attention. The main idea of RDHEI is that an image owner encrypts a cover image, and then a data hider embeds secret information in the encrypted image. With the information hiding key, a receiver can extract the embedded data from the hidden image; with the encryption key, the receiver reconstructs the original image. In this paper, we can embed data in the form of random bits or scanned documents. The proposed method takes full advantage of the spatial correlation in the original images to vacate the room for embedding information before image encryption. By jointly using Sudoku and Arnold chaos encryption, the encrypted images retain the vacated room. Before the data hiding phase, the secret information is preprocessed by a halftone, quadtree, and S-BOX transformation. The experimental results prove that the proposed method not only realizes high-capacity reversible data hiding in encrypted images but also reconstructs the original image completely.

1. Introduction

In the last few years, with the development of technologies for digital image processing, the transmission and exchange of images have become more and more convenient. At the same time, original images have begun to suffer from many security issues. Reversible data hiding (RDH) in cover images is a methodology that embeds secret messages into original images, such as law forensics, military, and medical images, in a reversible way such that the cover images can be completely recovered after the hiding data are extracted.

Reversible data hiding can be categorized into four major methods: lossless compression, difference expansion (DE), prediction error expansion (PEE), and histogram shifting (HS). The core idea of lossless compression [1–3] is that secret information is embedded into compressed images. Commonly, however, we cannot achieve high-capacity data hiding. Tian [4] proposed a difference expansion information hiding method that explored the vacant room in digital images. Compared to the traditional method, the data hiding capacity is significantly improved. The paper calculated all the differences between two adjacent pixels and embedded secret data into the values. DE was generalized by Alatter [5], in which n-1 bits can be embedded into a vector with n pixels. Thodi and Rodriguez [6] proposed an excellent extension of DE called prediction error expansion. PEE uses each pixel's prediction error instead of the pixel difference to hide information, to increase the hiding capacity beyond that of DE. A high-fidelity reversible data hiding method was proposed by Li et al. [7], in which a new prediction method called pixel value ordering (PVO) was combined with PEE. For each pixel block, the pixels are reordered into a pixel vector according to their values, and the secret data are embedded into two prediction errors, which correspond to the difference between the smallest pixel and the second smallest pixel and between the largest pixel and the second largest pixel. In [8], all k maximum pixels (or minimum pixels) were treated as a unit for information hiding. The previous method was based on performing PVO and PVO-k in a block-by-block manner, and generally, the capacity was small. In [9], a novel pixel-based PVO (PPVO) was proposed, in which the generated predictions were made in a pixel-by-pixel manner. Thus, the hiding capacity of PPVO was increased. The histogram shifting (HS) [10–12] method modifies the image histogram to embed secret information. It can produce high-quality marked images, while the hiding capacity of HS is limited.

With the development of cloud computing, the growth in information technology has led to serious security problems such as copying, hacking, or malicious usage of information. To ensure the secure transmission of digital images over the public network, two kinds of security techniques can be utilized: encryption and data hiding. RDHEI is an effective method for embedding secret information in the encrypted domain. There are three roles in RDHEI: the content owner, the data hider (cloud manager), and the receiver. The content owner encrypts the original image, the data hider embeds secret information into the encrypted image, and the receiver extracts the secret data and reconstructs the cover image. In recent years, RDHEI has received a lot of attention in the encryption stage [13–19] and the data hiding stage [20–24]. Until now, many RDHEI schemes have been proposed. On the basis of when the embedding space for additional data was created, i.e., before or after image encryption, embedding mechanisms of the current reversible data hiding schemes can be divided into two categories: vacating room after encryption (VRAE) and reserving room before encryption (RRBE). Generally, the RDHEI algorithm based on RRBE can embed a greater hiding capacity than that based on VRAE. However, an RDHEI scheme based on VRAE can reconstruct the original image without loss. Therefore, the former method has a relatively wider applicability than the second one.

In [13], the original image was divided into patches that were then represented according to an overcomplete dictionary via sparse coding. Because of regarding the patch as a whole, the number of coefficients was small. And thus a high-capacity room was available. The embedding capacity was 1.071 bpp and the quality of the embedded image was 40 dB. The average maximum embedding rate reached a factor of 1.7 as large as that of the previous best alternative scheme conducts. An RDHEI method based on the adaptive encoding strategy has been proposed [20]. During image encryption, block permutation and stream cipher encryption are applied to mask the original image. The permutation for blocks and pixels basically does not change the redundancy of the cover image. The embedding capacity was 1.8319 bpp and the quality of the embedded image was good. In [15], the paper constructed a new reversible data hiding scheme in an encrypted image with public key cryptography from difference expansion. In the previous RDHEI methods, the encrypted images rarely contained redundancy space. Thus, Liu and Pun [21] proposed a new novel RDHEI method based on reversible image reconstruction. The content owner rearranges the cover image to construct a redundancy image; simultaneously, the content of the cover image is made invisible. The rearranged image is used as an encrypted image. In 2019, Liu and Pun [23] proposed a novel reversible data hiding scheme in an encrypted image by redundant space transfer. To avoid destroying the majority of the redundant space, the paper designed an image encryption phase with three steps: disordering bit planes, disordering patches, and applying the Arnold transform. In order to reach better image visual quality. Wu and Sun [22] proposed two reversible data hiding schemes in encrypted images based on the prediction error: a separable method and a joint method. In [25], a lossless RDHEI method based on Chaos-Block was proposed. The paper used features of the pixel difference to embed more data than possible by other methods and carried out refinement with a single-level wavelet decomposition shifting technique to prevent image distortion problems. For the vacating room from the encrypted image without loss is difficult, a completely reversible data hiding in an encrypted image was proposed in [14]. This paper empties room by shifting the histogram of estimating errors of some pixels, which are estimated before encryption. In 2018, Qin et al. [16] proposed a high-capacity RDHEI method, in which the data hider first preprocesses all the encrypted patches by run-length coding and Huffman coding. Thus, a large amount of spare room is vacated to hide certain kinds of messages. In [17], the content owner preprocessed the cover image by the run-length coding and block-based MSB plane rearrangement schemes. The data hider can achieve high-capacity data hiding in an encrypted image. In the embedding phase [26], the encrypted image is adapted according to the error location map; thus, the receiver can extract secret data perfectly and reconstruct the cover image without any errors. But the algorithm cannot achieve high-capacity data hiding. In [18], before the image encryption, the content owner calculates the prediction values and marks the original image by Huffman coding. Then, it encrypts the cover image and embeds the label map into the encrypted image. The data hider embeds multibit data in each encrypted pixel by multi-MSB substitution according to the embedded label map. When the label map is large, however, there will be little redundant room. In [19], two RDHEI schemes were proposed, both of which are based on MSB prediction. In the first method, before encryption, the cover image is preprocessed. In the second method, the cover image is encrypted, and then prediction errors are embedded. The former method can embed more secret data into the encrypted image than the latter, but the latter method can reconstruct the original image without loss. To improve the security of the encrypted image and the quality of the decrypted image, Shu et al. [24] proposed an RDHEI based on neighborhood prediction using XOR-permutation encryption. In this method, the XOR-permutation is used to encrypt the cover image. Thus, the encrypted image can retain redundant room and statistical information.

In all cases, the presented RRBE and VRAE methods are not able to offer embedding rate and high reconstructed image quality simultaneously. Although the proposed methods could perform very well, they cannot be used for all images. It is necessary to propose a general method for high-capacity reversible data hiding in encrypted images. In this study, we present an efficient MSB method for high-capacity reversible data hiding in encrypted images based on information preprocessing. Because the Arnold transform [27] or Sudoku [28] permutation encrypts an image by displacing the pixel positions, the pixel values do not change. For this reason, it seems natural to transform reductant space from the original image to an encrypted image, as many image encryption methods. However, using the Arnold transform to process an image introduces a security risk. Because the transformation cycle of images with the same size is fixed. In this study, we use the Arnold transform and Sudoku permutation to reduce this risk. Normally, most existing methods cannot achieve a high embedding capacity (more than 1 bpp) because the redundant room is limited. In this study, the secret data are preprocessed by the halftone, quadtree [29], and DES [30] algorithms. When the secret data are scanned documents, we convert them from gray-level form to binary values by half-toning and then extract the content using the quadtree algorithm. Whether the original secret message is random bits or scanned documents, the message is compressed by S-BOX at the end. Each 6- bit data point can be compressed into 4 bits by S-BOX. For these reasons, we can achieve a high embedding capacity (more than 1 bpp) in the hiding phase. In the decoding phase, the receiver can extract the secret information and reconstruct the original image without loss using MSB prediction.

In this study, high-capacity data hiding and image reconstruction are concerned. For this problem, the schemes in earlier studies mainly involve the encrypt stage and data hiding stage. In the previous RDHEI methods, the encrypted image, processed by traditional standard encryption schemes, contains almost no redundant space. Some other compression coding methods are not conductive to image reconstruction. It is necessary to achieve high embedding capacity and high visual quality. Summarizing, the RDHEI schemes are on the basis of MSB prediction with a very high capacity. We will employ this method to handle the data hiding and image reconstruction operation.

The rest of the paper is organized as follows. Section 2 describes the proposed scheme in detail, including image encryption, data hiding, and data extraction and image recovery. Section 3 reports the experimental results and analysis. Finally, the conclusions are drawn and future work is proposed in Section 4.

2. Proposed Method

At present, few methods succeed in combining high embedding capacity (near 1 bpp) and high visual quality (greater than 50 db). An encrypted image is difficult to detect whether it contains secret data or not. Regarding the LSB (least significant bit) and MSB methods, their confidentiality is similar in the encryption stage, while the prediction of the MSB values is easier to carry out than that of the LSB values during decryption. To achieve high-capacity data hiding in encrypted images, we propose a method for high-capacity reversible data hiding in encrypted images based on information preprocessing. Sudoku and Arnold transformations both scramble the position of matrix elements without changing the size of the matrix. The encrypted image obtained through Sudoku and Arnold transformations has a good encryption effect and also retains redundant space. S-BOX is the core of the DES algorithm. It is the only nonlinear part of the algorithm and the key of the algorithm. The algorithm involves 8 S-BOX, each S-BOX with 6-bit input and 4-bit output. The first and sixth digits of the 6 digits in the S-BOX indicate the number of rows, and the middle four digits indicate the number of columns. Find the corresponding value by row number and column number. In this study, the S-BOX is used to preprocess the secret data, and the row number is transmitted through a special secure channel to achieve the effect of compression. The contributions of this study are summarized as follows. (1) We present a new encryption scheme with the Arnold and Sudoku transformations to transfer reductant spaces from the original image to an encrypted image. In this way, the redundant space is retained and the information of the cover image becomes invisible. (2) We preprocess the secret message before data hiding. The compressed data are embedded in the encrypted image. Thus, we can embed more than one bit per pixel. Figures 1(a)–1(c) show a flow chart of our scheme, including image encryption, data hiding, and data extraction and image recovery.

2.1. Image Encryption.

In the previous RDHEI methods, the encrypted image processed by traditional standard encryption schemes contains almost no redundant space. The encrypted image processed by the compression methods is difficult to be completely reconstructed. To avoid these problems, we design an Arnold transform [23] and Sudoku [28] matrix-based encryption method that preserve the redundant space and ensure security. The conventional schemes for data hiding are not flexible. There are not many chances to improve the security of data hiding. They are cracked and guessed easily. Therefore, we need more technologies for nonfixing and randomizing. With Arnold and Sudoku's independence and randomness, we enhance the unpredictability for an encrypted image and improve the security to be perceived.

In this stage, the cover image $I(M^*N)$ is processed by the Arnold transform to generate $I_A(M^*N)$. Then, the image $I_A(M^*N)$ is divided into m nonoverlapping blocks B_m sized 2^*2 and scanned, block by block, in scan line order, where $N_0 = M \times N/2 \times 2$ and $m = 1, 2, \ldots, N_0$. Each subblock is processed by Arnold transform with a different transform time. The Arnold transformed image I_A is divided into nonoverlapping blocks sized 2×2 and pixels, which are not in any subblocks without change. The divided blocks and their pixels are permuted by using the Sudoku matrix. In the previous RDHEI methods, the encrypted image processed by traditional standard encryption schemes contains almost no redundant space. To avoid this problem, we design an Arnold transform and Sudoku scrambling based encryption

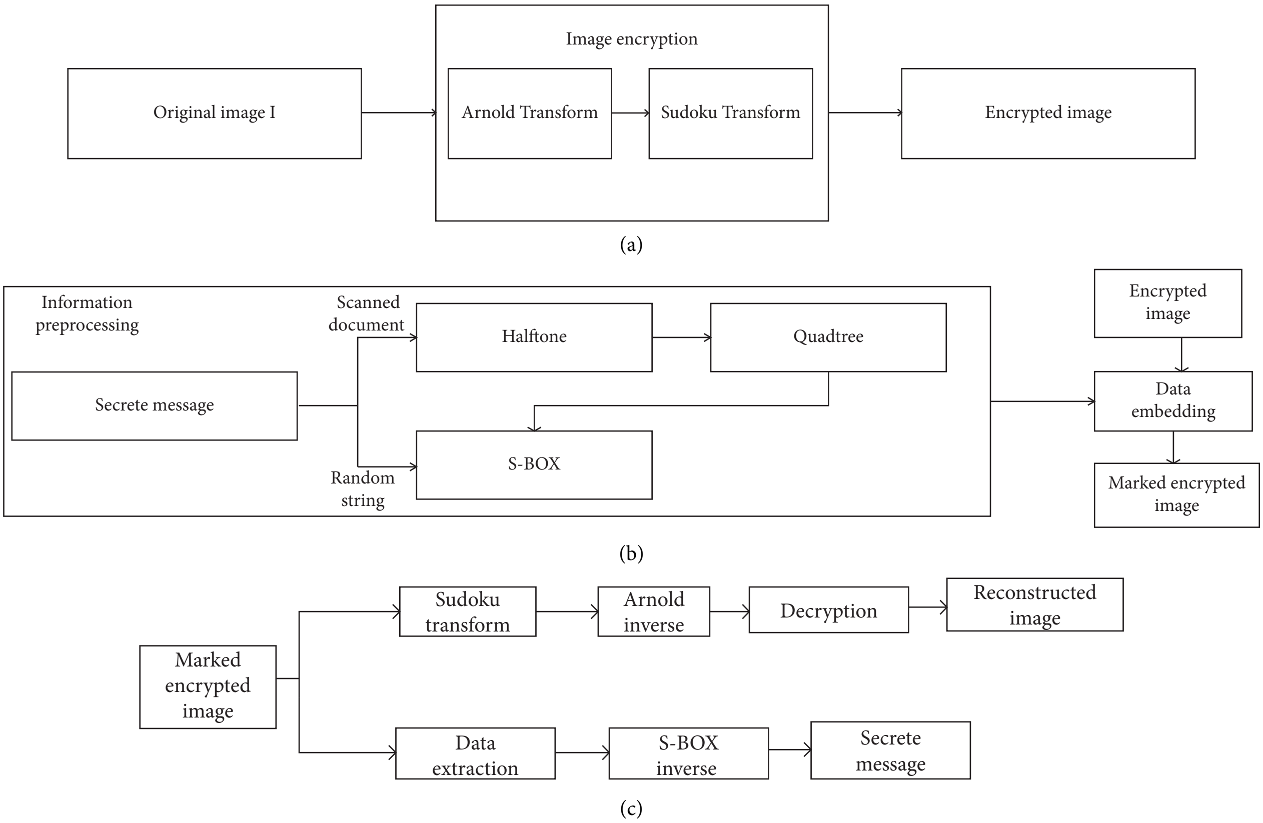


Figure 1: Framework of the proposed scheme. (a) Content owner side. (b) Data hider side. (c) Receiver side.

method. For the content owner, to generate encrypted images, there are two phases: (1) Arnold transform and (2) Sudoku permutation. Figures 2(a)–2(b) show a sample Sudoku puzzle and its solution. Figures 3(a)–3(c) are the lean image encryption process proposed in this study.

2.1.1. Prediction Error Detection. In this study, the compressed data are embedded by MSB substitution. Hence, the original MSB values are lost after the data hiding phase. During the decoding phase, to be able to predict pixels without any errors, the previous pixels are used to predict the current pixel value. Therefore, the content owner needs to analyze the original image content to detect all the possible prediction errors:

(1) Consider the current pixel value as $p(i, j)$ and its inverse value as $\mathrm{inv}(i, j)$. Here, $0 \le i \le M, 0 \le j \le N$, and $\mathrm{inv}(i, j) = (p(i, j) + 128)\mathrm{mod}256$

(2) Consider the average of the left and the top pixels as a predictor $\mathrm{pred}(i, j)$, which is considered as a predictor during the decoding step:

$$\mathrm{pred}(i, j) = \frac{p(i-1, j) + p(i, j-1)}{2}. \quad (1)$$

(3) Calculate the absolute difference between $\mathrm{pred}(i, j)$ and $p(i, j)$, as well as between in $v(i, j)$ and $p(i, j)$, denoted by e and e^{inv}, respectively.

$$\begin{cases} e = |\mathrm{pred}(i, j) - p(i, j)|, \\ e^{\mathrm{inv}} = |\mathrm{pred}(i, j) - \mathrm{inv}(i, j)|. \end{cases} \quad (2)$$

When $<e^{\mathrm{inv}}$, there is no prediction error. Otherwise, there is an error, and we store the value.

2.1.2. Sudoku Matrix. To preserve the redundant space of the original image, we introduce an image encryption method based on the Sudoku matrix and Arnold transform. It is known that Sudoku is a logical number fill game with the numbers 1 to 9 occurring exactly once in each subblock. Sudoku involves nine houses, and each house is divided into nine small squares. Our algorithm employs a Sudoku matrix and selects a subblock as our reference matrix. We use the reference matrix to permutate the pixel values. In the 3^*3 grid of the reference matrix, each value represents the position of the original pixel. In this study, we divide the original image into nonoverlapping blocks of 3^*3 in size. And then, according to the reference matrix, the pixels are permutated. Because the Sudoku matrix permutates only the position of each pixel and does not change its values, redundant space can be preserved. Furthermore, with the

3	1						8	5
			9		3			
9		5				3		7
8		4				1		6
			4		1			
6	9						7	3
	3		5		2		1	
			8		4			
	2		7		6		9	

(a)

3	1	2	6	4	7	9	8	5
7	8	6	9	5	3	2	4	1
9	4	5	1	2	8	3	6	7
8	5	4	3	7	9	1	2	6
2	7	3	4	6	1	8	5	9
6	9	1	2	8	5	4	7	3
4	3	7	5	9	2	6	1	8
5	6	9	8	1	4	7	3	2
1	2	8	7	3	6	5	9	4

(b)

Figure 2: A sample Sudoku puzzle and its solution. (a) A sample Sudoku puzzle. (b) Solution to the Sudoku puzzle in (a).

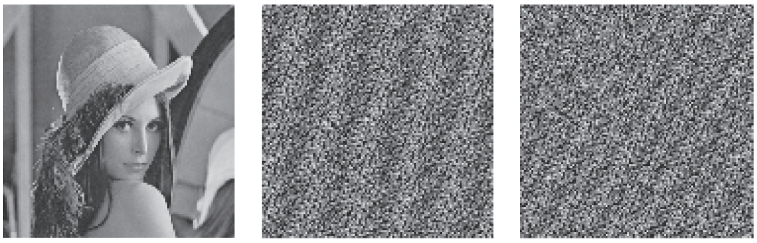

(a) (b) (c)

Figure 3: Illustration of our proposed method on the test image Lena (512*512): (a) original image; (b) Arnold transform image; (c) encrypted image.

property of the Sudoku matrix, the method can enhance the security of the encrypted image.

For example, a subblock matrix of an image is $\begin{bmatrix} 212 & 211 & 212 \\ 211 & 215 & 88 \\ 215 & 90 & 98 \end{bmatrix}$, and the 3*3 grid of the Sudoku puzzle is $\begin{bmatrix} 3 & 1 & 2 \\ 7 & 8 & 6 \\ 9 & 4 & 5 \end{bmatrix}$. The values of the 3*3 reference matrix represent the positions of the pixels. Our method scans the subblock of the image, from left to right and from top to bottom. According to the reference matrix, we can see that the first pixel is transferred to the second position and the third pixel is transferred to the first position, and so on. Finally, the subblock is permutated as $\begin{bmatrix} 212 & 212 & 211 \\ 215 & 90 & 88 \\ 98 & 211 & 215 \end{bmatrix}$.

2.1.3. Arnold Transform. In the 1960s, Arnold [27] first proposed the Arnold transform, in which the content owner encrypts an image by displacing pixel positions. The Arnold transform can be represented by a matrix as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod (N), \tag{3}$$

where $x, y \in \{0, 1, 2, \ldots, N-1\}$, the size of the original image is N^*N, $\begin{pmatrix} x \\ y \end{pmatrix}$ represent s the pixels of the original image, and $\begin{pmatrix} x' \\ y' \end{pmatrix}$ represents the pixels of the encrypted image. For an arbitrary pixel, the Arnold transform does not change its value, only its position. Thus, the Arnold transform preserves the redundant space from the original image and transfers it to the encrypted image. However, there is a security risk because the Arnold transform cycle of images with the same size is fixed. In order to reduce the risk, we divide the original image into subblocks and then adopt the Arnold transform to process each subblock with a different transform time.

We first use the Arnold transform to process the original image n_i times, in which n_i is less than the transform cycle T and the value of T is calculated by formula (4). Then, we divide the processed image into subblocks sized 3*3 and permute the pixels by using the Sudoku matrix. After completing the two steps, the original image I is encrypted, and the redundant space is transferred from the original image to the encrypted image I_{AS} . In this study, we set the Sudoku matrix and n_i as the secret keys.

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} (\mathrm{mod}N) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4}$$

2.2. Data Embedding. In the last few years, with the development of cloud computing, people have sought to embed more secret information into original images; at the same time, data privacy has become a major problem. In this study, the redundant space is transferred from the original image to the encrypted image. Thus, it is possible to achieve high-capacity reversible data hiding in an encrypted image. In many of the existing schemes, data hiding is carried out by MSB substitution and LSB substitution or the substitution of both. The previous methods cannot achieve high-capacity data hiding with most yielding results near 1 bpp. In the data hiding phase, we compress the secret data and then embed the compressed data into the encrypted image by MSB substitution. In many steganography schemes, any information is sent as random bits, while others make great efforts to embed and send scanned documents as secret data in a safe way. Any multimedia information can be converted into a binary string. Multimedia information contains a lot of redundancy. And it will be relatively large if directly converted into a binary string. Therefore, when the multimedia information is a secret message, eliminating redundancy will greatly improve image quality. In this study, when the secret data is a scanned document, we convert the scanned document from gray-level form to binary values by half-toning and then extract the parts with information by using the quadtree algorithm. When the message has a random bit form, the messages are compressed by S-BOX. For the messages are compressed before data hiding, we can easily increase the embedding capacity.

2.2.1. Halftone and Quadtree. The scanned document is converted to a binary image by the halftone method. It means that each 8-bit pixel is shown by 1 bit. Thus, the size is reduced by a factor of 8. Thus far, there are many methods for calculating the halftone image from a scanned document image, which are divided into three groups: error diffusion, dither, and iteration. In the present study, the method of error diffusion is used to calculate the halftone image. These halftone methods convert each pixel to 1 or 0 and the reverse halftone image calculation converts each 1 and 0 bit to an integer value between 0 and 255. The halftone image of a scanned document with the binary display can include signs such as text, images, and tables, which are shown with 0 bits and white backgrounds, as indicated by 1 bit. In this study, when the secret data are scanned documents, only the document content from the background is useful, not the background itself. Thus, we separate the content from the background and consider the content as secret data. In [29], an improved quadtree method was proposed, which had the ability to be applied to any image dimensions. Figures 4(a)–4(e) depict a scanned document image, halftone image, subrectangle image, content subrectangle image, and rectangle merging image. The quadtree scheme consists of the following steps:

(1) Scanned document images of any dimension are processed into N^*N images

(2) Each scanned document is converted to a halftone image by error diffusion and the size reduced by a factor of 8

(3) The halftone image is considered a rectangle. When the minimum width and height of the rectangle are larger than 1^*1, the rectangle should be segmented into subrectangles. This means that the rectangle is divided into four subrectangles, and this process is performed repeatedly on all subrectangles until there is no other subrectangle with division conditions

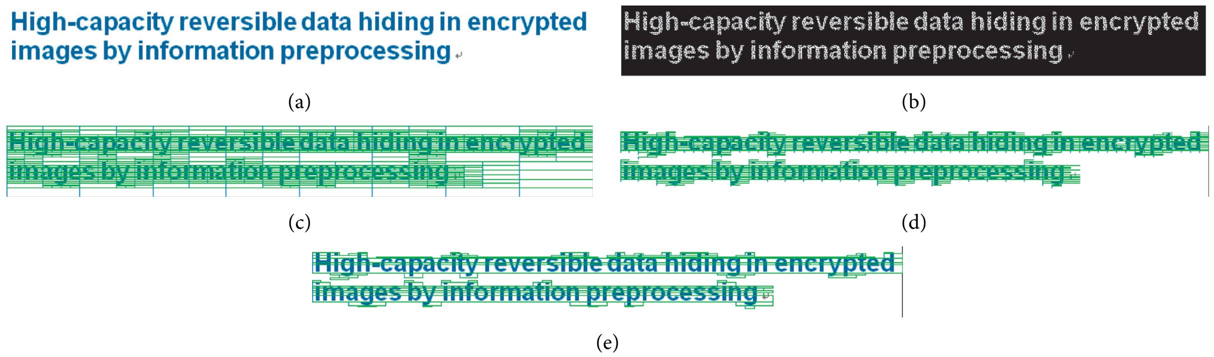
(4) Because there are the subrectangles without containing information, we only keep the content and coordinates of subrectangles that contain information. Normally, the number of these subrectangles can be high and 4 numbers are retained as coordinates for each subrectangle increase increasing the final amount of information

(5) All subrectangles that contain information are merged by scanning neighboring rectangles horizontally and vertically. Therefore, the number of merged rectangles can be small and 4 numbers can be retained as coordinates for each merged rectangle, reducing the size mentioned in step 4

(6) To ignore the 0s on the left side of a binary bit string, in our study, the decimal coding algorithm is used. Hence, we can read longer binary bitstreams and convert them to their equivalent decimal values. To achieve high-capacity data hiding, the data are processed by S-BOX and the binary bit strings are compressed by a factor of 1.5

In our study, the scanned document is processed by the above steps, yielding the rectangle merging image and the coordinates. We can see that the data were compressed very well. For example, we test the title of this paper in Figure 4, where (a) is the original scanned document image (24.3 KB, 24942 Byte), (b) is the halftone image, (c) is the subrectangles image, (d) is the content of the subrectangle image, and (e) is the rectangle merging image. As the result, the data are recorded by the text documents, and after step 5, we can obtain the data by data-bin.txt (192 KB, 197539 bytes). Through the decimal coding algorithm, we can obtain the data by data-dec.txt (35.6 KB, 36492Byte). Through the S-BOX operator, we can obtain the insert data by data-sbox.txt (23.7 KB, 24328 bytes). In this test, the secret data were compressed by a factor of 8.202.

2.2.2. S-BOX. S-BOX is the core of the DES algorithm. It is the only nonlinear part of the algorithm and the key to the


High-capacity reversible data hiding in encrypted images by information preprocessing
High-capacity reversible data hiding in encrypted images by information preprocessing
High-capacity reversible data hiding in encrypted images by information preprocessing
High-capacity reversible data hiding in encrypted images by information preprocessing
High-capacity reversible data hiding in encrypted images by information preprocessing

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

Figure 4: (a) Scanned document image; (b) halftone image; (c) subrectangle image; (d) content subrectangle image; (e) rectangle merging image.

security of the algorithm. The input of the S-BOX is 6 binary digits, and the output is 4 binary digits. According to the characteristics of the S-BOX, it can be used for packet compression and expansion. In this study, the S-BOX is adopted to preprocess the secret data, and the row number is stored and transmitted through a special secure channel. S-BOX is a simple substitution operation in which the input is a bitstream with each 6-bit group considered a set of data. And its output is a bitstream, but each set of data is converted from 6 bits to 4 bits. Table 1 presents substitution mapping. Assuming $A = a_1a_2a_3a_4a_5a_6$, let $k = a_2a_3a_4a_5$ and $h = a_1a_6$. In the row h and column k of S-BOX, we find a number $B = b_1b_2b_3b_4$. The value of B is in the range 0 to 15, proving that the substitution operator can compress the secret data from 6 to 4. In this study, we use all the numbers of these rows as secret keys.

In this study, before being hidden, the secret data were preprocessed. The secret data are a random bitstream or scanned document. When the secret data are a scanned document, the scanned document is converted into 4 numbers as coordinates for each merged rectangle and the content, and then, the data or the bitstreams are substituted by S-BOX. For example, regarding the bitstream 111000101010111000101010111000101010111000101010, first, each 6-bit group is treated as a set. Thus, we obtain 8 groups of data: 111000, 101010, 111000, 101010, 111000, 101010, 111000, and 101010. Second, each group can be substituted by S-BOX. Thus, we obtain 8 substitution groups of data, i.e., .0011, 0110, 0011, 0110, 0011, 0110, 0011, and 0110, and the numbers of rows 10, 10, 10, 10, 10, 10, 10, and 10 are kept as secret keys.

2.2.3. Data Embedding. In the data hiding stage, the data hider first preprocesses the secret message by a halftone, quadtree, and S-BOX. Then, the preprocessed data are embedded in the encrypted image without knowing the encryption key. Pixels of the encrypted image are scanned from left to right and then from top to bottom, and the preprocessed data are embedded into MSB planes by MSB substitution:

$$p_{\text{em}}(i, j) = b_k \times 128 + (p_e(i, j))\text{mod}128. \tag{5}$$

2.3. Receiver. In the decoding phase, the receiver receives the marked encrypted image p_{em}, and then the receiver can conduct data extraction and image recovery, which include two scenarios: (1) if the receiver has only a data hiding key, the secret message can be extracted; (2) if the receiver has the encryption key and the data hiding key, the secret data and cover image can be recovered with no error.

2.3.1. Data Extraction. The pixels from the marked encrypted image are scanned from left to right, and then from top to bottom, and the MSB of each pixel is extracted to recover the reprocessed secret message:

$$b_k = p_{\text{em}}(i, j)/128, \tag{6}$$

where $0 \le k < M \times N$ and refers to the index of the recovered bit in the preprocessed secret message.

For example, regarding the bitstream 00110110001101100011011000110110, first, each 4-bit group is regarded as a set. Thus, we obtain 8 groups of data: 0011, 0110, 0011, 0110, 0011, 0110, 0011, and 0110. Second, according to the numbers of rows, each group can be substituted by S-BOX. Thus, we can obtain 8 substitution groups of data: 111000, 101010, 111000, 101010, 111000, 101010, 111000, and 101010. When the secret data are a scanned document, the bit string is first operated by S-BOX, and then the coordinates and the content of the rectangles can be obtained. Finally, we recover the scanned document.

2.3.2. Image Recovery. In this study, three steps are adopted to realize image reconstruction:

① The marked encrypted image p_{em} is adjusted by a Sudoku sequence to obtain the reconstructed image p_{e1}.

In this study, the marked encrypted image is divided into subblocks sized 3*3 and scanned from left to right and then from top to bottom. Pixels in each subblock are adjusted by a Sudoku matrix. For example, a subblock matrix of the marked encrypted image is $\begin{bmatrix} 212 & 212 & 211 \\ 215 & 90 & 88 \\ 98 & 211 & 215 \end{bmatrix}$, and the 3*3 grid of the

Table 1: S-BOX.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Sudoku puzzle is $\begin{bmatrix} 3 & 1 & 2 \\ 7 & 8 & 6 \\ 9 & 4 & 5 \end{bmatrix}$. According to the values of the Sudoku grid, the pixels of the marked encrypted image are transferred to $\begin{bmatrix} 212 & 211 & 212 \\ 211 & 215 & 88 \\ 215 & 90 & 98 \end{bmatrix}$.

② The reconstructed p_{e1} image is operated on by an Arnold inverse transformation:

$$\begin{pmatrix} x^{,\prime} \\ y^{,\prime} \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \bmod (N), \tag{7}$$

where $\begin{pmatrix} x' \\ y' \end{pmatrix}$ represents the pixels of thepe1 image and $\begin{pmatrix} x'' \\ y'' \end{pmatrix}$ represents thepixels of the reconstructed p_{e2} image .

③ The MSB value is calculated:

(a) The pixel value is considered for MSB = 0 and MSB = 1. And then, we calculate the differences between each of these two values and pred(i, j). The two differences are recorded as e_1 and e_2:

$$\begin{cases} e_1 = \left| \mathrm{pred}(i, j) - p(i, j)^{\mathrm{MSB}=0} \right|, \\ e_2 = \left| \mathrm{pred}(i, j) - p(i, j)^{\mathrm{MSB}=1} \right|. \end{cases} \tag{8}$$

$$p(i, j) = \begin{cases} p(i, j)^{\mathrm{MSB}=0}, & \text{if } e_1 < e_2, \\ p(i, j)^{\mathrm{MSB}=1}, & \text{else.} \end{cases} \tag{9}$$

(b) The smaller value between e_1 and gives the reconstructed pixel value:

3. Experimental Results and Comparisons

In this section, we present the results obtained by using our method with the high-capacity reversible data hiding in encrypted images based on information preprocessing. To evaluate the performance of our proposed scheme, we first applied our study on the original image of 512 × 512 pixels of the USC-SIPI image database in this experiment. PSNR (Peak Signal-to-Noise Ratio) was used to evaluate the visual quality degradation of the encrypted image produced by our study. The high PSNR value indicated that the visual artifact of the encrypted image was imperceptible to human visual sensitivity. SSIM (Structural Similarity Index Measurement) was used to evaluate the similarity of two images. When the SSIM value was 1, it indicates that the reconstructed image and the cover image were the same. Section 3.1 lists a full example of our method and shows the results in the test images. We perform a statistical analysis to test the capacity and the visual security of our study. Finally, in Section 3.2, we compare our approach with related methods and discuss its efficiency.

For data hiding in encrypted images, we need to measure different performances which are the number of incorrectly extracted bits, the payload, and the recovered image quality after message extraction. We are interested in discovering a general method to improve the embedding capacity for all images and to discover the best trade-off between all the above parameters.

3.1. Detailed Example for the Proposed Study. In our study, the scanned document of [31] was used as secret information (each page was scanned in the scale of 1700*2388). In Table 2, the size of the eight scanned documents and corresponding sizes was listed. We first applied our study on the test images, sized 512×512, from the USC-SIPI image database. In Figure 5, for the same scanned document, the results of the average compression ratio for five different thresholds with the rectangular size of 1 × 1, 4 × 4, 8 × 8, 16 × 16, and 32 × 32 were 8.001296, 4.573963, 2.992956, 2.051653, and 1.665751, respectively. Eight different pages were used as secret information, and Lena was adopted as the cover image. In Figure 6, for the Lena image, the results of embedding capacity for five different thresholds with the rectangular size of 1 × 1, 4 × 4, 8 × 8, 16 × 16, and 32 × 32 were 442082 bits, 770898 bits, 1178221 bits, 1721599 bits, and 2119225 bits, respectively. For the same document, when the minimum rectangle size was 1 × 1, we can see that the actual embedding capacity was the highest and the mean hiding capacity of the test image was 1.6865 bpp. Hence, in this study, the minimum size of rectangles of 1 × 1 was adopted. The experiment is shown in Figures 7 and 8, in which the embedding rate is 1.4483 bpp when the secret data are a bitstream. And we preprocess the secret data with S-BOX. Sometimes the secret data are a scanned document, in which case, the embedding rate is 7.714 bpp. In this study, we preprocess the scanned documents with the halftone, quadtree, and S-BOX technologies. For the test image, all

Table 2: Eight scanned documents and corresponding sizes.

Page number	Size (B)
1	308002
2	507020
3	524171
4	467128
5	419473
6	436539
7	492878
8	378691

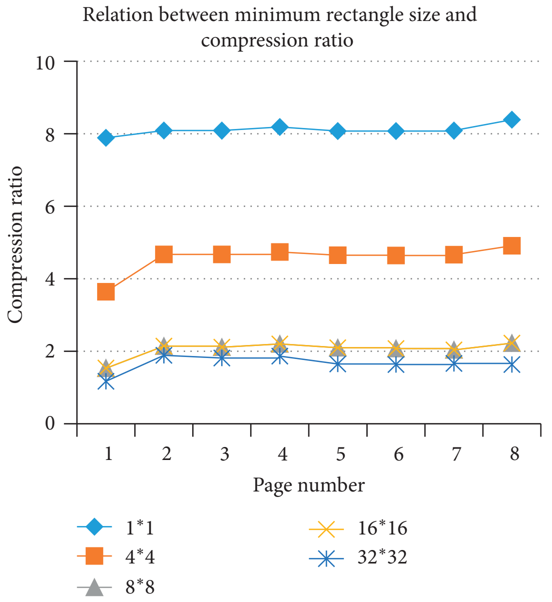


Figure 5: Relation between minimum rectangle size and compression ratio.

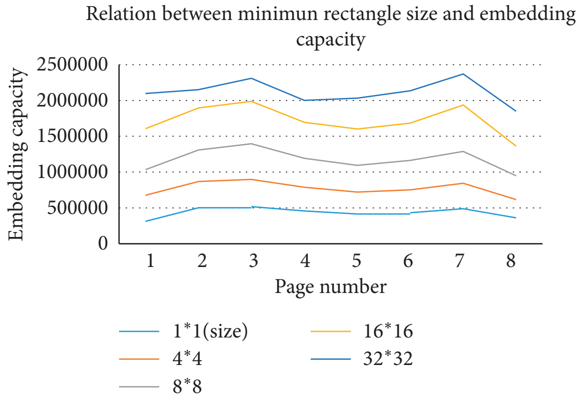


Figure 6: Relation between minimum rectangle size and embedding capacity.

pixels are correctly recovered (PSNR = ∞, SSIM = 1). In Table 2, the results of applying the proposed method to the test images are shown. The PSNR tends toward +∞ and the embedding rate by more than 1 bpp. However, as the amount of embedding is greater, the quality of the reconstructed image will decline.

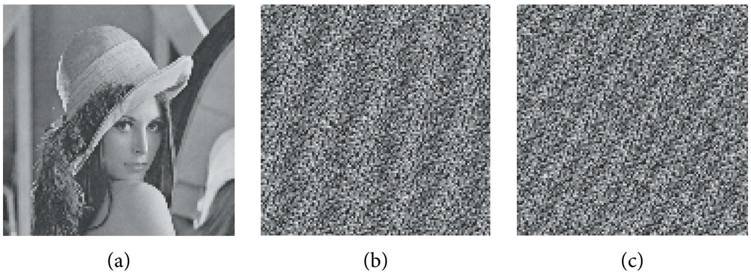
3.2. Security Analysis of the Proposed Study. We analyze the security of our study through the keyspace parameter, which involves the total number of possible combinations of an encryption key. Usually, a large keyspace can effectively ensure that encrypted contents are not accessible by unauthorized users. According to the detailed explanation in Section 2, the key includes two parts: (1) the total number of different Sudoku solutions was 6.67×10^{21} and (2) the second part is the Arnold transform parameters E_A.

The number of possible combinations of E_A can be calculated by $\mathrm{KS}_A = ((\log_2 M) - 1) \times (T - 1)^{M/2\times N/2}$.

Therefore, the keyspace of our study is $6.67 \times 10^{21} \times \mathrm{KS}_A$. Supposing a $512 \times 512 \times 8$ image to be encrypted by our study, the value of T is set to 3, and then the keyspace of our study is $6.67 \times 10^{21} \times 8 \times 2^{65536.}$ The keyspace of our study is sufficiently large to resist many kinds of brute force attacks. In addition, the secret information extracted by the tamper is garbled. The message is the real message after the inverse transformation of the S-BOX.

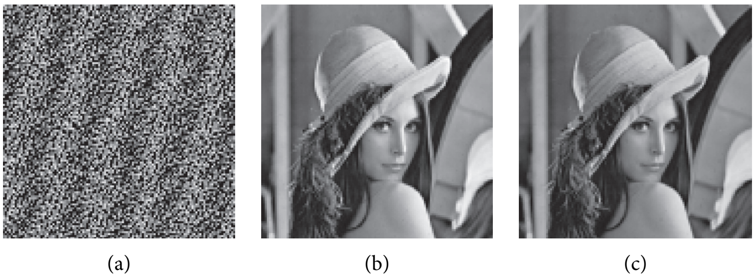
3.3. Comparisons with Related Studies and Discussions. We applied our method to two kinds of different secret data and present the detailed results obtained on the random bits and scanned documents. In Table 3, when the secret data are random bits, we make some comparisons between our study and five existing ones which are [13, 14, 19, 22, 25]. In this study, we use the four test images presented in Figure 6; the results are listed in Table 4. In [19], they can totally reverse all the images, and the SSIM is equal to 1 and the PSNR tends toward +∞. In [13, 14, 22, 25], they cannot totally reverse all the cover images, and the embedding capacity of the four methods is all less than 1. With our proposed approach, we achieve results of 1.3824 bpp. The SSIM is equal to 1, and the PSNR tends toward +∞. In Table 5, we can see that the secret data is a scanned document, and in our study, the true embedding data are coordinates for each merged rectangle and then compressed by S-BOX. Thus, the embedding capacity is higher. We achieve results of 7.714 bpp. The SSIM is equal to 1 and the PSNR tends toward +∞. In conclusion, in addition to being error-free during secret data extraction, our study allows us to have a good trade-off between the hiding capacity and the recovered image quality after data extraction. From the security point of view, the statistical analysis shows that there is no information about the content of the cover image in the marked encrypted version. Most importantly, we used the S-BOX substitution, which invalidates the information even if an attacker extracts it.

Secret information preprocessing is a contribution of this paper and can be extended to existing advanced processing algorithms [13, 14, 19, 22, 25]. The previous hiding algorithms directly embed the secret message into the image. Applying the preprocessing module to the previous algorithms can further increase the embedding capacity. To realize high-capacity information embedding, it is worth storing and transmitting the row numbers of S-BOX exclusively.



(a) (b) (c)

Figure 7: Illustration of our proposed method on the test image Lena (512*512): (a) original image; (b) encrypted image; (c) marked encrypted image.



(a) (b) (c)

Figure 8: Illustration of our proposed method on the test image Lena (512*512): (a) marked encrypted image; (b) decrypted image; (c) recovered image.

Table 3: Comparisons between [13, 14, 19, 22, 25] and our study.

Test image	Methods	Embedding rate (bpp)	PSNR (dB)
Lena	[13]	0.7556	38.38
	[22]	0.1563	+∞
	[25]	0.4	30.01
	[14]	0.0667	55.56
	[19]	0.9641	+∞
	Ours (bitstream)	1.4483	+∞
Airplane	[13]	0.95	42.22
	[22]	0.1563	60.17
	[25]	0.3722	30.01
	[14]	0.05	55.56
	[19]	0.9889	+∞
	Ours (bitstream)	1.4833	+∞
Lake	[22]	0.1563	54.84
	[19]	0.9839	+∞
	Ours (bitstream)	1.4762	+∞
Baboon	[22]	0.1563	40.57
	[19]	0.7478	+∞
	Ours (bitstream)	1.1217	+∞

Table 4: Payload measurements (in bpp) on test images.

Test image	Secret data	Embedding rate (bpp)	PSNR (dB)
Lena	Bitstream	1.4483	+∞
	Scanned document	7.7141	+∞
Airplane	Bitstream	1.4833	+∞
	Scanned document	7.9925	+∞
Lake	Bitstream	1.4762	+∞
	Scanned document	7.8725	+∞

Table 4: Continued.

Test image	Secret data	Embedding rate (bpp)	PSNR (dB)
Baboon	Bitstream	1.1217	+∞
	Scanned document	5.9834	+∞
Man	Bitstream	1.4682	+∞
	Scanned document	7.8317	+∞
Crowd	Bitstream	1.47825	+∞
	Scanned document	7.8853	+∞

Table 5: Comparisons between [13, 14, 19, 22, 25] and our study.

Test image	Methods	Embedding rate (bpp)	PSNR (dB)
Lena	[13]	0.7556	38.38
	[22]	0.1563	+∞
	[25]	0.4	30.01
	[14]	0.0667	55.56
	[19]	0.9641	+∞
	Ours (scanned document)	7.7141	+∞
Airplane	[13]	0.95	42.22
	[22]	0.1563	60.17
	[25]	0.3722	30.01
	[14]	0.05	55.56
	[19]	0.9889	+∞
	Ours (scanned document)	7.9925	+∞
Lake	[22]	0.1563	54.84
	[19]	0.9839	+∞
	Ours (scanned document)	7.8725	+∞
Baboon	[22]	0.1563	40.57
	[19]	0.7478	+∞
	Ours (scanned document)	5.9834	+∞

4. Conclusions

In our study, we proposed an efficient MSB method for high-capacity data hiding in encrypted images based on information preprocessing that outperforms the previous state-of-the-art methods. We encrypt the original image by an Arnold and a Sudoku transformation. Thus, we retain room for redundancy. To achieve high-capacity reversible data hiding, we preprocess the secret data before the embedding operator. We can see that the S-BOX phase sacrifices memory for embedding capacity. In this study, the S-BOX and Sudoku matrix are saved and sent to the receiver in a separate encrypted channel. In future work, we are interested in hiding more secret data; for example, we can perform multiple secret data compression transformations before embedding data, and the second MSB of each pixel can be used to enlarge the amount of embedded data.

Data Availability

The data used to support the fndings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no financial conflicts of interest.

Authors' Contributions

Xi-Yan Li and Xia-Bing Zhou contributed equally to this work. Xiyan Li designed the experiments and wrote the paper. Xia-Bing Zhou edited the English text of a draft of this manuscript and carried out the experiments and data collection. Qinglei Zhou, Shijing Han, and Zheng Liu performed statistical analyses and data interpretation. Zheng Liu reviewed and revised the manuscript.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (61572444).

References

[1] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Proceedings of the SPIE Proceedings of Security Watermarking Multimedia contents III*, vol. 4314, pp. 197–208, San Jose, CA, USA, January 2001.

[2] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in *Proceedings of the SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents IV*, vol. 4657, pp. 572–583, San Jose, CA, USA, April 2002.

[3] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[5] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[6] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans on Image Processing*, vol. 16, no. 3, pp. 21–730, 2007.

[7] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Processing*, vol. 93, no. 1, pp. 198–205, 2013.

[8] Bo Qu, X. Li, Y. Zhao et al., "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Processing:Image Communication*, vol. 29, no. 7, pp. 760–772, 2014.

[9] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Processing*, vol. 111, pp. 249–260, 2015.

[10] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[11] C.-C. Lin and N.-L. Hsueh, "A lossless data hiding scheme based on three-pixel block differences," *Pattern Recognition*, vol. 41, no. 4, pp. 1415–1425, 2008.

[12] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.

[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] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, pp. 118–127, 2014.

[15] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226–233, 2015.

[16] C. Qin, Z. He, X. Luo, and J. Dong, "Reversible data hiding in encrypted image with separable capability and high embedding capacity," *Information Sciences*, vol. 465, pp. 285–304, 2018.

[17] K. Chen and C.-C. Chang, "High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 334–344, 2019.

[18] Y. Xiang, Z. Yin, and X. Zhang, "Reversible data hiding in encrypted images based on MSB prediction and huffman coding," 2019, http://arxiv.org/abs/1812.09499.

[19] P. Puteaux and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1670–1681, 2018.

[20] Y. Fu, P. Kong, H. Yao, Z. Tang, and C. Qin, "Effective reversible data hiding in encrypted image with adaptive encoding strategy," *Information Sciences*, vol. 494, pp. 21–36, 2019.

[21] Z.-L. Liu and C.-M. Pun, "Reversible image reconstruction for reversible data hiding in encrypted images," *Signal Processing*, vol. 161, pp. 50–62, 2019.

[22] X. Wu and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," *Signal Processing*, vol. 104, pp. 387–400, 2014.

[23] Z.-L. Liu and C.-M. Pun, "Reversible data-hiding in encrypted images by redundant space transfer," *Information Sciences*, vol. 433-434, pp. 188–203, 2018.

[24] Y. Shu, C. Fan, and H. He, "Reversible data hiding in encrypted image based on neighborhood prediction using XOR-permutation encryption," *Journal of Computer Research and Development*, vol. 55, no. 6, pp. 1211–1221, 2018.

[25] M. Bartwal and R. Bharti, "Lossless and reversible data hiding in encrypted images with public key cryptography," *Rice*, vol. 10, pp. 127–134, 2017.

[26] P. Puteaux and W. Puech, "High-capacity reversible data hiding in encrypted images using MSB prediction," in *Proceedings of the IPTA 2016*, Oulu, Finland, December 2016.

[27] V. I. Arnold and A. Avez, *Ergodic Problems of Classical Mechanics*, Benjamin press, New York, NY, USA, 1968.

[28] J.-C. Cheng, W.-C. Kuo, and B.-R. Su, "Data-hiding based on Sudoku and generalized exploiting modification direction," *Journal of Electronic Science and Technology*, vol. 16, no. 2, pp. 123–129, 2018.

[29] S. H. Soleymani and A. H. Taherinia, "High capacity image data hiding of scanned text documents using improved quadtree," 2018, http://arxiv.org/abs/1803.11286v1.

[30] L. Chen, *Modern Cryptography*, Science Press, Beijing, China, 2008.

[31] N.-I. Wu and M.-S. Hwang, "A novel LSB data hiding scheme with the lowest distortion," *The Imaging Science Journal*, vol. 65, no. 6, pp. 371–378, 2017.