applied sciences

Article

Data Hiding and Authentication Scheme for Medical Images Using Double POB

Fang Ren [1,2], Xuan Shi [1,2,*], Enya Tang [1] and Mengmeng Zeng [1]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; renfang_81@163.com (F.R.); tangenya1122@163.com (E.T.); zengmengmiao@163.com (M.Z.)
[2] National Engineering Research Center for Secured Wireless, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
* Correspondence: xuann1118@163.com; Tel.: +86-13992448337

Abstract: To protect the security of medical images and to improve the embedding ability of data in encrypted medical images, this paper proposes a permutation ordered binary (POB) number system-based hiding and authentication scheme for medical images, which includes three parts: image preprocessing, double hiding, and information extraction and lossless recovery. In the image preprocessing and double hiding phase, firstly, the region of significance (ROS) of the original medical image is segmented into a region of interest (ROI) and a region of non-interest (RONI). Then, the bit plane of the ROI and RONI are separated and cross-reorganization to obtain two new Share images. After the two new Share images are compressed, the images are encrypted to generate two encrypted shares. Finally, the embedding of secret data and attaching of authentication bits in each of these two encrypted shares was performed using the POB algorithm. In the information extraction and lossless recovery phase, the POBN algorithm is first used to extract the authentication bits to realize image tamper detection; then, the embedded secret message is extracted, and the original medical image is recovered. The method proposed in this research performs better in data embedding and lossless recovery, as demonstrated by experiments.

Keywords: data hiding; medical images; POB number system; lossless recovery

Citation: Ren, F.; Shi, X.; Tang, E.; Zeng, M. Data Hiding and Authentication Scheme for Medical Images Using Double POB. *Appl. Sci.* **2024**, *14*, 2664. https://doi.org/10.3390/app14062664

Academic Editor: Antonino Pietropaolo

Received: 19 February 2024
Revised: 16 March 2024
Accepted: 19 March 2024
Published: 21 March 2024

1. Introduction

With the swift growth of internet technology and network infrastructure, electronic medical care is also developing rapidly. Nowadays, medical data are mainly maintained electronically as electronic patient records (EPRs); EPR data include medical images, health record data, patient history, etc., that need to be transmitted. Medical images have highly strict image quality criteria because they are one of the most significant tools used by doctors to diagnose illness. Any small change may lead to an incorrect diagnosis or even serious medical accidents. To secure the confidentiality, dependability, and availability of medical images, many data-hiding algorithms and image encryption algorithms have been proposed. Furthermore, to prevent tampering with the integrity of encrypted images, image restoration algorithms have become a popular research topic.

Data-hiding technology is often used to protect patients' privacy information in telemedicine applications. This technology refers to the process of attaching data information that needs to be transmitted to the carrier image. There are already many data-hiding algorithms based on medical images, Priyanka et al. [1] introduced a hybrid image watermarking scheme based on the separation of ROI and RONI, and medical data and least significant bits (LSBs) data from the ROI were attached to the RONI. A high-volume steganography algorithm based on medical images was proposed by Wang et al. [2]. Thakur et al. [3] introduced a robust telemedicine application watermarking technology that includes singular value decomposition and combines Discrete Wavelet Transformation

(DWT) and Discrete Cosine Transform (DCT) transforms. A new adaptive reversible watermarking scheme applied to medical images was proposed by Lee [4]; the algorithm extends the estimation error of adjacent pixels to embed the watermark and can automatically segment the target region and background region of a medical image.

Although the visual and embedding quality of the image can be guaranteed by data-hiding techniques, privacy protection requirements may not be satisfied by simply embedding the hidden data into the original image. Therefore, reversible data hiding in encrypted images (RDHEI) technology has emerged as a prospective method. Chen et al. [5] separated the encrypted image into blocks to make room for hiding secret messages by compressing the difference between pixels. Wang and He [6] proposed a scheme to compress the block using the uniform most significant bit (MSB) of the pixels in the block, which achieved good results, but this scheme is not suitable for complex images. Fu et al. [7] determined the embeddable block by analyzing the position of the MSB layer in the encrypted image. To make space for embedding data, the MSB layer of the embeddable block needs to be compressed in an adaptive way. In order to preserve the redundancy in the encryption block, Liu et al. [8] adopted the classical XOR method to encrypt the block and used the redundancy matrix existing in the encrypted block to generate the available space that could accommodate the secret message. Gao [9] split the image through blocks for encryption and compressed it using the data hider to free up space for embedding the encrypted secret message. To guarantee the quality of the decrypted image, Qin et al. [10] separated the image into a texture block and a smooth block, and they just inserted additional data into the smooth block. Nowadays, most block-based RDHEI schemes deal with data hiding and encryption separately. As a result, it is required and critical to investigate a method for synchronizing data embedding and encryption in order to reduce the threats of potentially insecure encryption systems.

RDHEI technology can only ensure the safety of medical images; it cannot guarantee the dependability and availability of medical images. Therefore, in the field of integrity and authentication of medical images, Singh et al. [11,12] proposed a technique to detect and locate the tampered area, which also has the ability to recover tampering by embedding authentication bits into each block’s three LSBs, and recovery bits are attached to mapping blocks; this technique restores the tampered area with 50% accuracy. Tiwari et al. [13] use two watermarking methods to detect and locate real-time tampering in medical images and other common images. A highly efficient fragile watermarking strategy was proposed by Gull et al. [14]. A fragile watermarking technique was proposed by Shehab et al. [15], who replaced an image’s LSB with authentication and self-recovery bits in order to identify tampering and recover tampered regions. Parah et al. [16] used fragile watermarks and block checksums to detect and locate tampering. Additionally, in [17], Gao et al. improved ROI contrast without adding distortion and used it for tamper detection.

The permutation ordered binary (POB) number system was introduced recently by Sreekumar et al. [18], aiming to construct a novel secret sharing method. It is applied to medical images to enhance security and protect the integrity of medical images. Compared with some existing algorithms, the POB number system has been further developed and applied in tamper detection, lossless recovery, bit plane compression, and so on. Ren et al. [19] tried to use POB as a solution for re-encrypting the contents of the image during the data hiding process, but this scheme had a low embedding capacity. Using a POB number system on a cloud server, Singh and Xiang [20,21] presented a safe method for detecting and recovering from image tampering. Liu et al. [22] used image tamper detection and lossless recovery, combined with POB digital system to perform neighborhood refinement and watermark refinement. The results of the experiment demonstrated that, under some circumstances, the method performed well, but there were still some problems in recovering large-scale tampering. To solve such problems, Li et al. [23] proposed a scheme for medical image authentication recovery based on the POB algorithm. Experimental findings and theoretical research have demonstrated that this strategy allows the lossless recovery of original images under different tampering attacks, but this scheme can only achieve simple authentication and still has defects in carrying additional data.

Targeting the aforementioned inadequacies, this research proposes a medical image hiding and authentication algorithm based on double POB system. The scheme not only realizes the authentication of the medical image but can also transmit additional secret data during the authentication. Firstly, the region of significance (ROS) part of the original medical image is separated into an ROI and an RONI by a segmentation algorithm, and then these two regions are separated and cross-reorganized to generate two shares so as to prevent the high-bit sensitive pixels in the ROI from being tampered with. Then, the two Share images were compressed, and the compressed data were repeatedly placed in the plane with the same size as the original image; then, the plane was encrypted using a hyperchaotic Lorenz system. Finally, the double POB number system was used to complete the data hiding and authentication. After successful authentication, the embedded secret message is extracted, and the original medical image is recovered. If the data are tampered with, the original image can still be recovered completely by repeating the data placement.

The primary contributions of this research are summarized as follows:

(1) We utilized a double POB digital system to simultaneously achieve data hiding and the authentication of medical images. The embedding capacity of data hiding is large, and the process is reversible; the authentication is based on the pixel level, which is very sensitive to any small tampering and meets the needs of medical images.

(2) We propose a method based on bit plane separation and cross-reorganization to protect sensitive information in medical images. The bit plane separation method is based on image segmentation, and the reorganization method protects the high-bit sensitive pixels in the ROI.

(3) We propose a method for medical image tampering recovery by compressed data repeated filling. In the authentication stage, if the authentication fails due to tampering, the data of the untampered area can be directly used for tampering recovery.

The structure of the rest of this paper is as follows. Section 2 describes the basic theory related to the proposed scheme. In Section 3, the hiding and authentication process based on double POB system in medical images is detailed. The experimental results and analysis are given in Section 4. Finally, this paper is concluded in Section 5.

2. Preliminaries

This section contains some preliminary information about image segmentation, the POB number system, the hyper-chaos Lorenz system, and Huffman coding.

2.1. Image Segmentation

There are various medical image segmentation methods widely used at present, and the OTSU algorithm is a classical threshold segmentation method.

The OTSU algorithm, also referred to as the maximal variance between clusters, is an algorithm that utilizes the image histogram to determine the optimal global threshold. The advantages include simplicity, a fast calculation speed, and independence from image brightness and contrast. The basic principle of the OTSU algorithm is as follows. Firstly, it automatically finds a threshold to divide the image and distinguish the foreground region from the background region. It divides the histogram into two groups at a certain threshold. The threshold is determined by maximizing the variance between these two groups of histograms, thus achieving the optimal segmentation effect.

Specifically, assume that a grayscale image has a range of gray values of $[0, L-1]$; the pixel number of gray value i is n_i, the number of total pixels is denoted as N, and each gray value has a probability of p_i. The first step involves obtaining a threshold value, denoted as k. Subsequently, this value is utilized to partition the gray values into two distinct groups. C_0 represents the set of gray values in pixels that are less than or equal to k, while C_0 denotes the set of gray values in pixels greater than k. The probabilities of C_0 and C_1 are denoted as m_0, m_1, respectively, so $m_0 + m_1 = 1$; the average values of C_0 and C_1

are denoted as w_0, w_1, respectively; the whole image's average gray value is denoted by w; and $w(k)$ is the average gray value when the threshold is k. The specific expression is shown in Equation (1).

$$\begin{cases} N = \sum_{i=0}^{L-1} n_i \\ p_i = \frac{n_i}{N} \\ w = \sum_{i=0}^{L-1} ip_i \\ w(k) = \sum_{i=0}^{k} ip_i \end{cases} \quad \begin{cases} m_0 = \sum_{i=0}^{k} p_i = m(k) \\ m_1 = \sum_{i=k+1}^{L-1} p_i = 1 - m(k) \\ w_0 = \sum_{i=0}^{k} \frac{ip_i}{m_0} = \frac{w(k)}{m(k)} \\ w_1 = \sum_{i=k+1}^{L-1} \frac{ip_i}{m_1} = \frac{w-w(k)}{1-m(k)} \end{cases} \tag{1}$$

The grayscale average of all samples is:

$$w = m_0 w_0 + m_1 w_1 \tag{2}$$

The between-class variance of two sets of gray values can be obtained with the following equation:

$$\begin{aligned} \sigma^2(k) &= m_0(w_0 - w)^2 + m_1(w_1 - w)^2 \\ &= m_0 m_1 (w_1 - w_0)^2 \\ &= \frac{[wm(k)-w(k)]^2}{m(k)[1-m(k)]} \end{aligned} \tag{3}$$

Traversing the gray level $[0,\ L-1]$, the inter-class variance under each threshold k is calculated in turn, and the k value that can make $\sigma^2(k)$ obtain the maximum value is the optimal threshold.

2.2. POB Number System

The POB number system was proposed by Sreekumar and Sundar [18]. The system is denoted POB (n, r), where $n \geq r$ and both n and r are nonnegative integers. In this context, r stands for the number of 1s in a string, and n stands for the number of bits. In the POB number system, the range of P(A) is $0,\ 1, \cdots \binom{n}{r} - 1$, where $A = a_{j-1}a_{j-2}\cdots a_0$.

The following can be used to obtain P(A):

$$P(A) = \sum_{j=0}^{n-1} a_j \binom{j}{v_j} \tag{4}$$

where $v_j = \sum_{j=0}^{n-1} a_j$.

The POB number in binary form can be calculated by the above equation to obtain the matching POB value and vice versa.

For example, the 10-bit binary string 1011001010, denoted as POB (10, 5), has a POB value that can be calculated by Equation (4).

$$\begin{aligned} lP(A) &= a_0 \times \binom{0}{0} + a_1 \times \binom{1}{1} + \cdots\cdots + a_8 \times \binom{8}{4} + a_9 \times \binom{9}{5} \\ &= 0 + 1 + 0 + \binom{3}{2} + 0 + 0 + \binom{6}{3} + \binom{7}{4} + 0 + \binom{9}{5} \\ &= 185 \end{aligned}$$

The obtained POB value of 185 can be converted to the corresponding 8-bit binary, that is, 10111001. This result has a crucial impact on some research based on data hiding. It can simultaneously perform lossless compression and re-encryption in the process of data hiding.

Similarly, 010111011 is denoted as POB (9, 6), and its corresponding POB value is 10, whereas a POB value of 28 would be 010101101 for POB (9, 5). Since the images used in this paper are all 8-bit images, and both secret information and authentication bits are embedded with two bits, the 10-bit POB (10, r) string in each operation can be converted into an equivalent 8-bit decimal POB value, which ranges from 0 to 251.

2.3. Hyperchaotic Lorenz System

In this section, hyperchaotic Lorenz systems are introduced. The method proposed by Wang et al. [24] is superior in performance, which adds a nonlinear controller w to the Lorenz system to produce hyperchaotic behavior. The system is expressed as follows:

$$\begin{cases} \dot{x} = a(y - x) + w, \\ \dot{y} = cx - y - xz, \\ \dot{z} = xy - bz, \\ \dot{w} = -yz + rw. \end{cases} \tag{5}$$

where a, b, c and r are system parameters, and x, y, z and w are state variables. When a = 10, b = 8/3, c = 28, and $-1.52 \leq r \leq -0.06$, the system represents a hyper-chaos state. Hyperchaotic systems have a higher value in secure communication than ordinary chaotic systems.

2.4. Huffman Coding

Huffman coding is a lossless encoding method in statistics. Its principle is based on the frequency of data (pixels in an image), using fewer bits to encode data with a higher frequency, which can reduce the overall coding length to achieve the purpose of data compression. Huffman coding produces the lowest number of redundant codes, has low computational complexity, and is easy to implement. Huffman coding has been effectively applied to text, image, video compression, and so on. For a more detailed description of Huffman coding, refer to [25].

3. The Proposed Scheme

This part proposes a new scheme based on double POB system for medical image hiding and authentication. In the use of medical images, users are more concerned with the information in the brighter areas, which is called the ROS (region of significance), and the information in the darker areas is redundant in most cases. Therefore, in order to minimize the degradation of medical image quality, the image owner performs image preprocessing on the ROS for operation in the first place and then inserts secret data and authentication bits to generate two shares, which are then transmitted to the receiver. After the receiver authenticates the image, they extract the ciphertext information and complete the lossless recovery. The work of the image owner is shown in Figure 1.

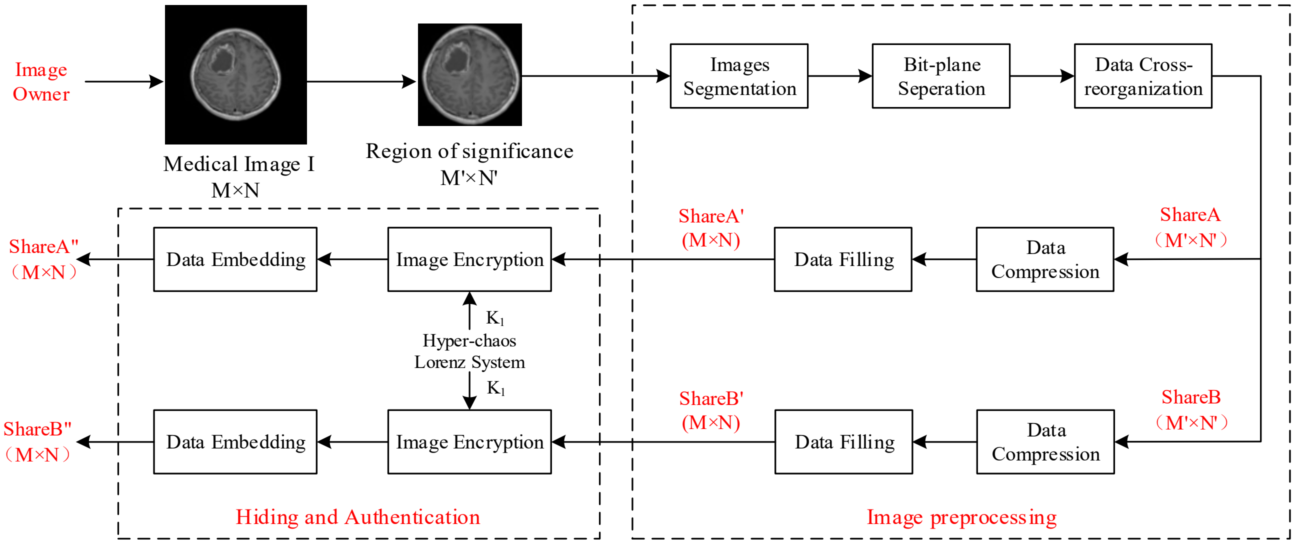


Figure 1. The work of the image owner.

3.1. Image Preprocessing

The image owner performs the pre-processing of the ROS of the extract, including image segmentation, bit plane separation, data cross-reorganization, data compression, and data filling.

3.1.1. Image Segmentation and Bit Plane Separation Cross-Reorganization

To better protect the important part of the ROS, the original ROS is first segmented, and the best threshold is selected by the OTSU algorithm. The part over the threshold is the ROI, and the part under the threshold is the RONI. After segmentation, the two regions start to undergo bit plane separation reorganization. Firstly, the high five bits of the ROI and the low three bits of the RONI are combined and paired into a new 8-bit image called ShareA. Then, the low three bits of the ROI and the high five bits of the RONI are combined and paired into a new 8-bit image called ShareB. This results in ShareB containing less important pixel information and a more important ShareA. ShareB is prioritized over ShareA in the subsequent information embedding process, protecting the ROI of the medical image.

The specifics of the data cross-reorganization process are illustrated in Figure 2, and Equation (6) explains it.

$$\begin{aligned} ShareA(i) &= [High5Plane_{ROI}(i) + Low3Plane_{RONI}(i)] \\ ShareB(i) &= [High5Plane_{RONI}(i) + Low3Plane_{ROI}(i)] \end{aligned} \tag{6}$$

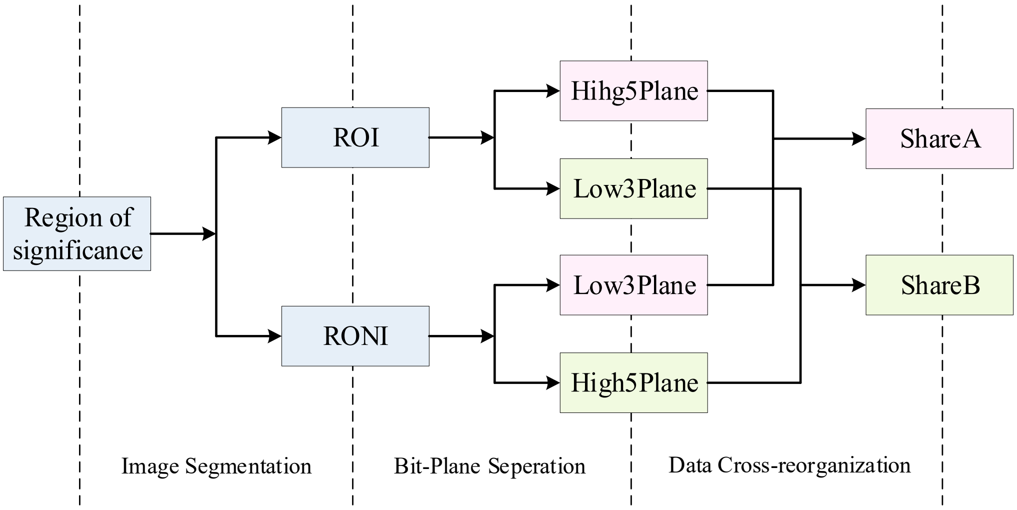


Figure 2. The process of image segmentation and bit plane separation reorganization.

3.1.2. Data Compression and Data Filling

In this part, ShareA is first divided into 8 × 8 nonoverlapping blocks, and then the Huffman coding algorithm is used to compress each block. Finally, each block's compressed data are stored in C, where the compressed data's length is n, and then it is repeatedly filled in a Share image of 512 × 512 as the size of the original medical image, which needs to be filled repeatedly $(512 \times 512)/n$ times in order to facilitate image recovery. The same data compression operations are performed for ShareB.

We give a concrete example: if the original medical image size is 512 × 512, then it has 262,144 pixels. If the size of the extracted ROS is 352 × 352, it has 123,904 pixels, and the extracted ROS pixels are reduced by almost half; then, ShareA is compressed in blocks, and the number of pixels after compression is about 30,621. In order to obtain an image of size 512 × 512, the compressed data must be filled at least $\lfloor 262,144/30,621 \rfloor = 8$ times. The processes of data compression and data filling are shown in Figure 3.

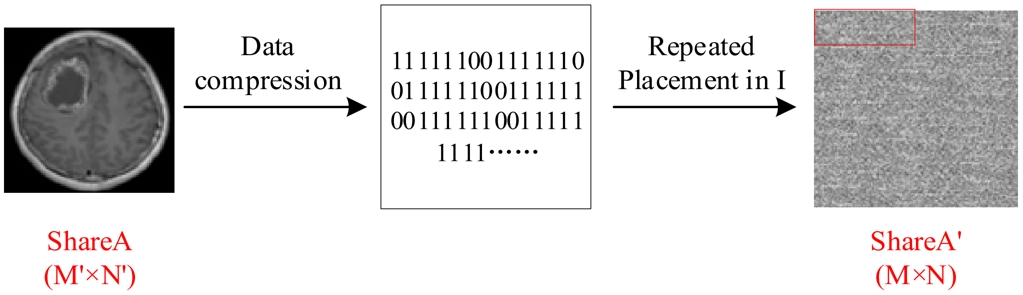


Figure 3. Processes of data compression and data filling.

3.2. Double Hiding

The image owner performs image encryption and data embedding on the preprocessed image.

3.2.1. Image Encryption

Two new Share images, namely ShareA′ and ShareB′, are obtained after data compression and data filling are completed. To encrypt the two new Share images, the hyperchaotic Lorentz system generates random numbers with the same size as the image, where the key is recorded as K_1, and the same key is used for encryption and decryption. Then, the generated random numbers are used for XOR encryption so that two encrypted ShareA′ and encrypted ShareB′ images with the same size as the original medical image are obtained, and the subsequent secret message embedding based on POB and the authentication information embedding based on POB are carried out in these two encrypted Shares.

3.2.2. Double Embedding Based on POB

The POB algorithm is used to embed sensitive messages (such as patient identity, condition, medical history, etc.) into the encrypted Share, which has the function of embedding and re-encryption synchronization. Since encrypted ShareB′ contains less important pixel information than encrypted ShareA′, in order to protect the information in sensitive areas from being damaged, the former has a higher priority than the latter when embedding secret messages, so the secret message is first embedded in encrypted ShareB′.

Secret message embedding based on POB concrete steps is as follows:

Step 1: Firstly, it traverses the pixels of encrypted ShareB′ to determine whether the length of the secret message exceeds the capacity of encrypted ShareB′. If not, the secret message is directly embedded, and the second and third steps are repeated until all information is embedded. If it exceeds the capacity, step 4 is completed after completing step 2 and step 3.

Step 2: The secret message $w_0, w_1 \in \{0, 1\}$ is selected and embed it in each pixel of encrypted ShareB′ to make the pixel turn to a ten-bit binary number.

Step 3: Through the POB number system, the pixels embedded with secret information in encrypted ShareB′ are compressed from ten-bit binary numbers to eight-bit binary numbers, and new pixels containing secret information are generated.

Step 4: If the capacity of encrypted ShareB′ is not enough, the embedding is continued in encrypted ShareA′ using the methods in steps 2 and 3.

After the secret message is embedded, the encrypted image containing the secret message is obtained, and the POB is used again to embed the authentication bit in the two encrypted images containing the secret information.

The authentication information embedding based on POB concrete steps is as follows:

Step 1: Each 8-bit pixel of the encrypted image that contains the secret message is used to calculate two authentication bits, and they are then attached to the 8-bit pixel.

Step 2: The first one in the authentication bit is denoted as the number of 1s in the first four digits of the eight pixels, and the second one is denoted as the number of 1s in the last four digits, which is denoted as 1 when odd and 0 when even.

Step 3: Finally, the ten-bit pixels embedded with authentication bits are converted into eight-bit POB values through the POB number system, and ShareA″ and ShareB″ are generated.

As shown in Figure 4, pixels 169 and 98 are selected as examples to carry out double embedding based on the POB system.

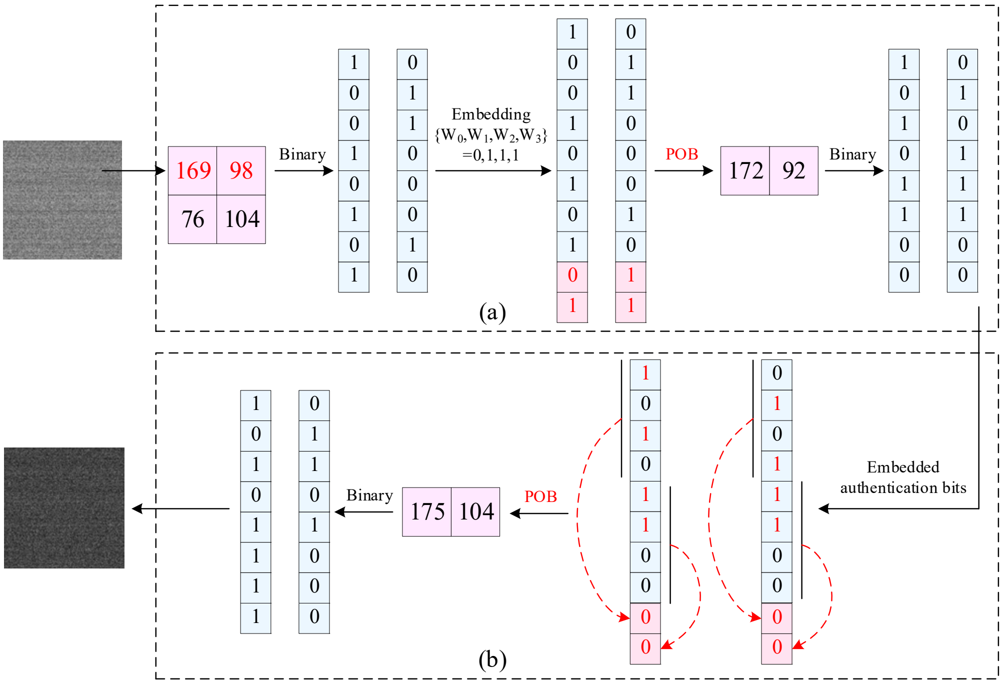


Figure 4. Double embedding based on the POB system. (**a**) Secret message embedding based on the POB system. (**b**) Authentication information embedding based on the POB system.

The double embedding based on the POB algorithm not only greatly increases the capacity of data hiding but also enhances the security of images. This is because the method can automatically re-encrypt the currently processed pixels during data hiding and authentication bit embedding, thus destroying the local correlation preserved by the design of the cryptosystem, and the encrypted image, after embedding, is more difficult to understand.

3.3. Information Extraction and Lossless Recovery

In order to obtain a patient's information for remote treatment, the recipient needs to extract the embedded secret message from the encrypted shared image they receive. The medical image must be protected using the lossless recovery method for data recovery. Figure 5 illustrates the information extraction and lossless recovery scheme using the example of ShareA″. Firstly, an authentication bit is extracted from ShareA″ for authentication bit detection. If it is not tampered with, the receiver extracts the secret message and completes image recovery. If it is detected that the Share has been tampered with, only the tampering recovery operation is performed. The same operation is performed for ShareB″.

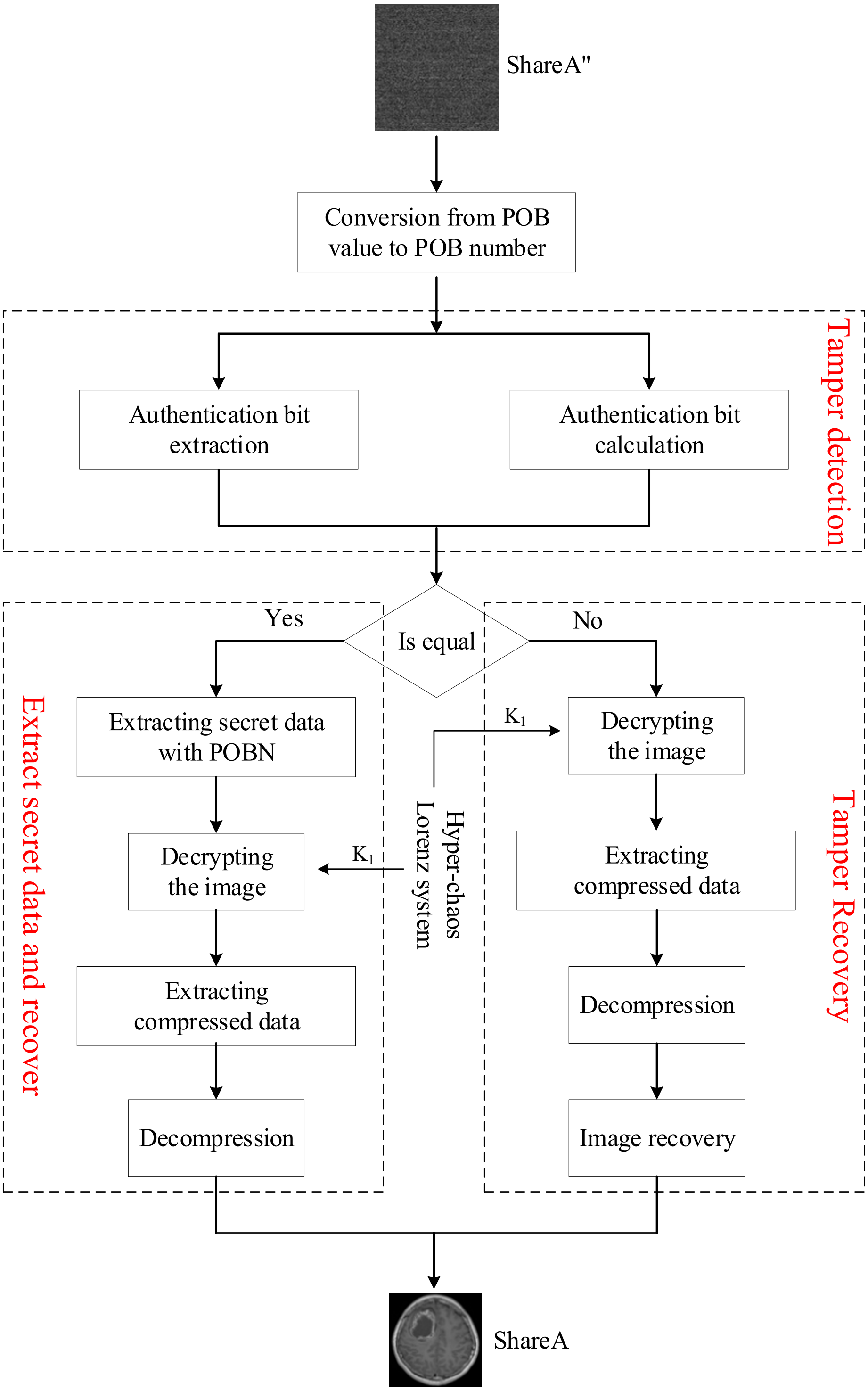

ShareA"
Conversion from POB value to POB number
Tamper detection
Authentication bit extraction
Authentication bit calculation
Is equal
Yes
No
Extract secret data and recover
Extracting secret data with POBN
Decrypting the image
Extracting compressed data
Decompression
K_1
Hyper-chaos Lorenz system
K_1
Tamper Recovery
Decrypting the image
Extracting compressed data
Decompression
Image recovery
ShareA

Figure 5. Process flow chart of extracting secret information and image recovery.

3.3.1. POBN Algorithm

The receiver adopts two inverse POB algorithms in order to recover the received Share image, as the two received shares were formed in accordance with the POB number system. Refer to Algorithm 1, also known as the POBN algorithm, for further information. Two rounds of the POBN algorithm are performed. In order to extract the authentication bit from each pixel, the POBN algorithm is first used to convert the POB value to the corresponding POB number. The authentication bit in the 10-bit data is then erased if it is found to be identical to the authentication bit determined by the statistical technique, indicating that the image has not been tampered with. In the second execution of the POBN algorithm execution, each pixel's hidden information is extracted, the POB value is converted to a ten-POB number after the identity bit is removed, the embedded information is extracted from each POB number, the ten-bit data are converted to eight-bit data, and the encrypted ShareA′ and encrypted ShareB′ are recovered.

Algorithm 1 Converts the POB value to the corresponding POB number

```
Input: $n$, $r$, and $v$, where $r \leq n$ and $0 \leq v \leq \binom{n}{r} - 1$.
Output: the POB number $A = a_{j-1}a_{j-2}\cdots a_0$.
(i) Let $j = n$ and temp = $v$.
(ii) For $k = r$ down to 1, perform the following:
  1. Repeat {
  2.   $j = j - 1$;
  3.   $p = \binom{j}{k}$;
  4.   if (temp $\geq p$)
  5.       temp = temp $- p$;
  6.       $a_j = 1$;
  7.   else $a_j = 0$;
  8.    } until $(a_j = 1)$;
(iii) If ($j > 0$)
      for $k = j - 1$ down to 0, perform the following:
        $a_k = 0$;
      end.
(iv) Return A.
```

3.3.2. Secret Information Extraction and Image Authentication

The same key used for the encryption is used to decrypt the two encrypted Shares that are recovered. The compressed data are extracted and then decompressed to obtain ShareA and ShareB following decryption. The detailed steps are as follows, using ShareA″ as an example:

Step 1: If the extracted and calculated authentication bits are the same, the process starts from the third step; if they differ, it proves that the image has been tampered with, and the process starts from the second step.

Step 2: The tampered regions are located by the comparison results of the authentication bits, and only the third, fourth, and fifth steps are performed on the regions that are not tampered with.

Step 3: After the authentication bit is removed, the POBN algorithm is used to convert the 8-bit POB value to the 10-bit POB number, and the embedded secret information is extracted.

Step 4: The random number generated by the hyperchaotic Lorentz system is used for XOR decryption, and the key used for decryption is identical to the encryption key. After decryption, an image composed of repeated filling with compressed data is obtained.

Step 5: After extracting the compressed data from the filled image and decompressing the data, ShareA, composed of the reorganized data, is recovered.

The same steps are performed for ShareB″ until the embedded secret information is extracted and ShareB is recovered.

3.3.3. Image Reorganization and Recovery

The obtained ShareA and ShareB undergo bit plane separation, and the data are reorganized. To obtain the ROI data, the high five bits of ShareA and the low three bits of ShareB are extracted, and then they are combined. Then, extract the low three bits of ShareA and the high five bits of ShareB and combine them to obtain the RONI data. Finally, combine the ROI and RONI data to recover the ROS. Figure 6 depicts the detailed steps involved.

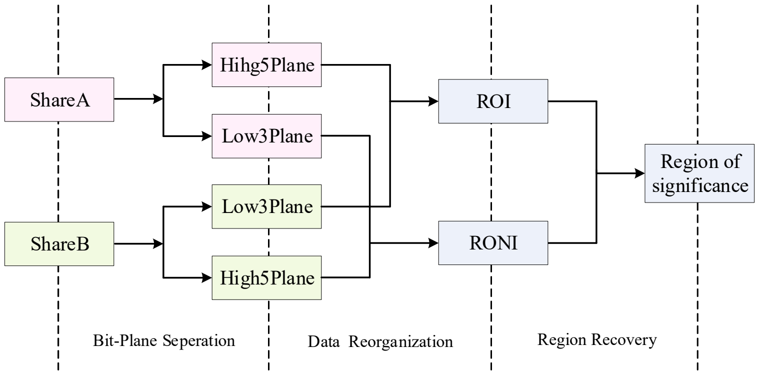


Figure 6. The process of data reorganization and region recovery.

4. Analysis of Simulation Result

In this part, Python 3.6 is used to simulate the proposed scheme and evaluate its performance. The medical image set was chosen for experimental testing to provide a better estimation of the algorithm's performance. As shown in Figure 7, eight medical images of different sizes, different body parts, and different types were selected for simulations to verify the effectiveness of the scheme.

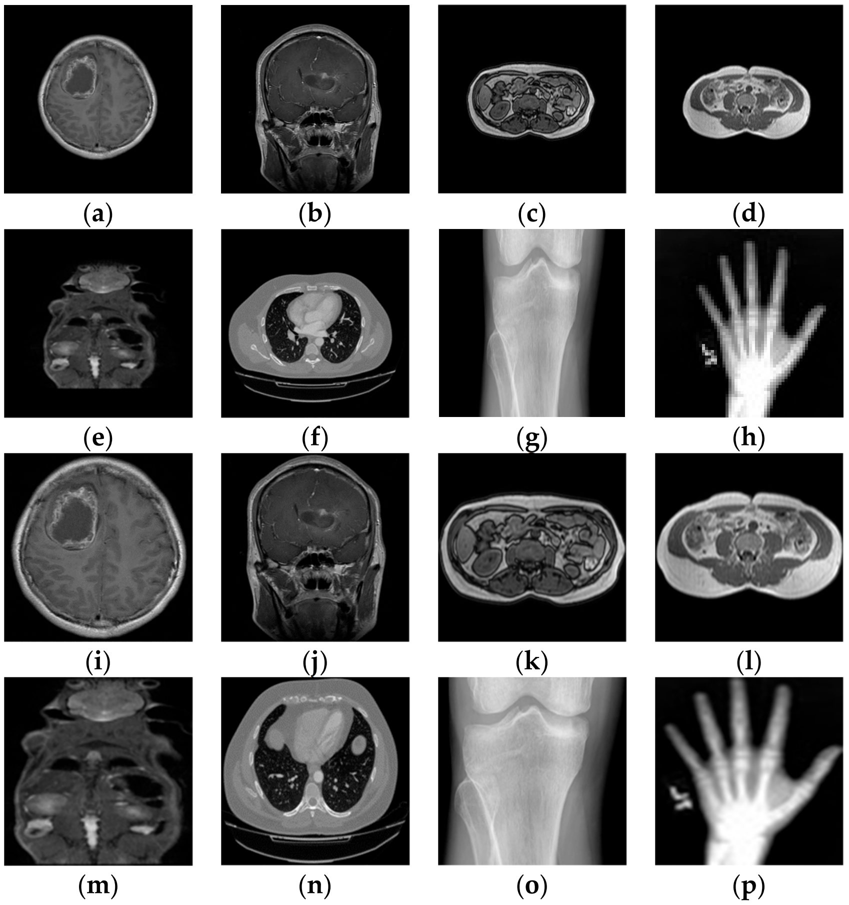


Figure 7. The standard medical images used for testing. (**a**) Brain 01 MRI (512 × 512); (**b**) Brain 02 MRI (512 × 512); (**c**) Pulmonary 01 CT (256 × 256); (**d**) Pulmonary 02 DICOM (256 × 256); (**e**) Pancreas MRI (180 × 180); (**f**) Chest CT (256 × 256); (**g**) Knee X-ray (300 × 162); (**h**) Hand X-ray (64 × 64); (**i**–**p**) The ROS region of extracted. (**i**) Brain 01 (352 × 352); (**j**) Brain 02 (464 × 464); (**k**) Pulmonary 01 (176 × 176); (**l**) Pulmonary 02 (176 × 176); (**m**) Pancreas (120 × 120); (**n**) Chest (228 × 188); (**o**) Knee (212 × 162); (**p**) Hand X-ray (46 × 60).

4.1. Encryption Performance

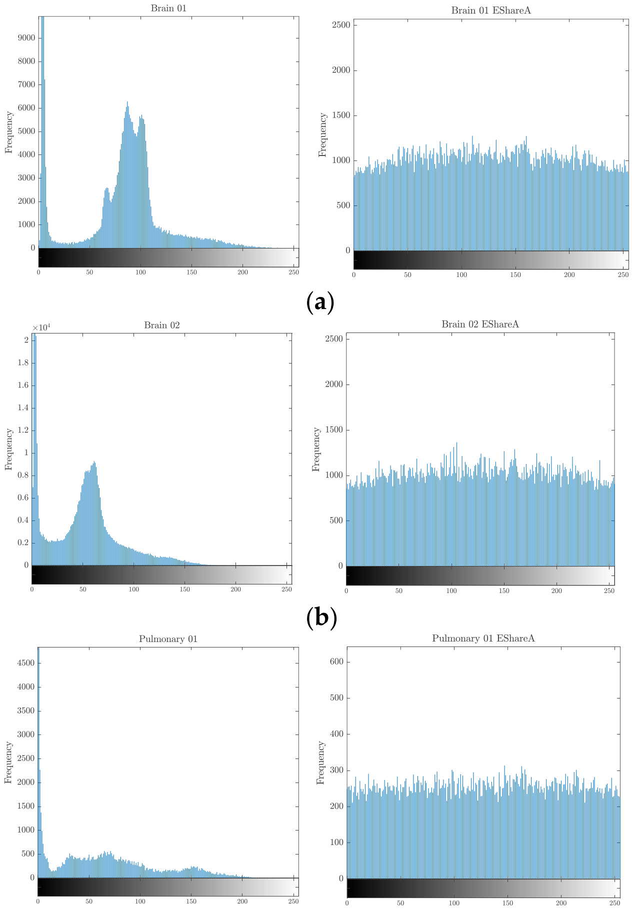
In this section, ShareA is selected for testing. We analyze the proposed scheme's security through the key space, the gray histogram, and the correlation of adjacent pixels.

4.1.1. Key Space Analysis

For the encryption method in this paper, the four initial values of the hyperchaotic Lorentz system are $K = \{x_0, y_0, z_0, w_0\}$, which are double precision numbers, so the size of the key spaces are $(10^{16})^4 \approx 2^{213}$. It can be seen that the key space of the hyperchaotic Lorentz system is very large, which makes it very difficult to crack the key. In addition, when using the POB for hiding and authentication, converts a 10-bit POB (10, r) string to an equivalent POB value, it can range from 0 to $\begin{pmatrix} n \\ r \end{pmatrix} - 1$. For example, when n is 10 and r is 4, there are 210 possible POB values. Thus, the probability of successfully analyzing a Share image for an image of size $M \times N$ is $\left(\frac{1}{210}\right)^{M \times N}$.

4.1.2. Gray Histogram Analysis

The gray histogram describes the number of pixels of different brightness levels in an image, which is an important feature in statistical analysis. An ideal encrypted image histogram should be uniform and flat. Figure 8 shows the gray histograms of the original and encrypted images. The encrypted image's histogram appears to be very different from the original image's histogram, with a quite uniform distribution. Therefore, the encryption scheme can resist statistical analysis.

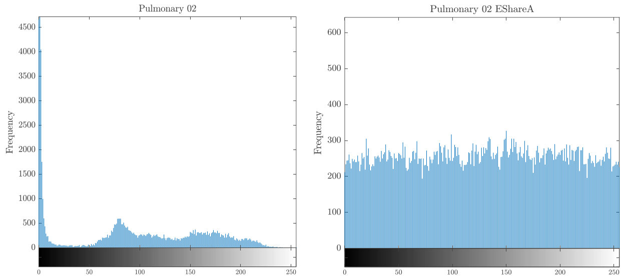


(**a**)

(**b**)

(**c**)

Figure 8. *Cont.*



(**d**)

Figure 8. The gray histograms of the original and encrypted images. (**a**) Brain 01; (**b**) Brain 02; (**c**) Pulmonary 01; (**d**) Pulmonary 02.

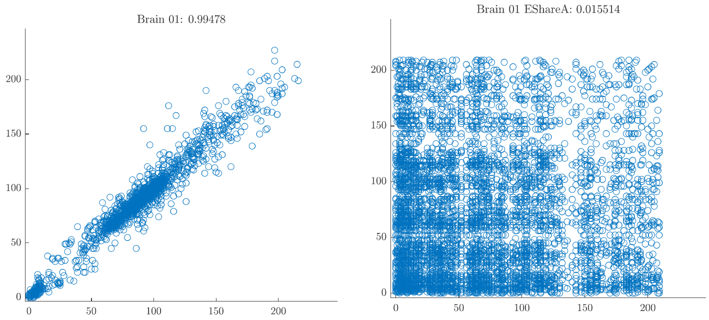
4.1.3. Correlation of Adjacent Pixels Analysis

The degree of correlation between adjacent pixels in an image is represented by the correlation of adjacent pixels. A meaningful image has a high degree of correlation between adjacent pixels in all directions. When encrypting an image, breaking strong pixel correlation is an important part of the image encryption algorithm. An effective image encryption algorithm should attempt to minimize the correlation between adjacent pixels and try to achieve zero correlation. Usually, the encrypted image has a low correlation coefficient, so it looks like a noisy image. The correlation of adjacent pixels can be obtained with Equation (7).

$$C = \frac{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N} x_i\right)\left(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N} x_i\right)^2 \times \sum_{i=1}^{N}\left(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i\right)^2}} \tag{7}$$

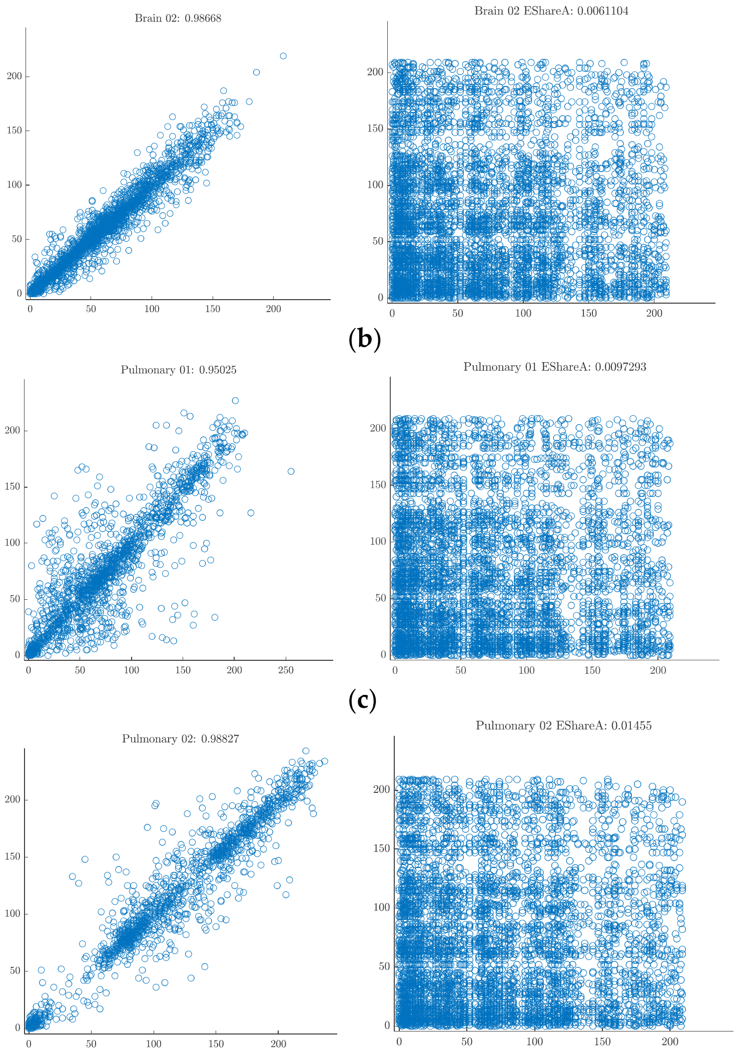
From the image's grayscale pixel value matrix, N pairs of adjacent pixels are selected at random, where the pixel values of two adjacent pixels are represented by the parameters x_i and y_i.

In order to verify that the POB has the property of re-encryption, we analyze and compare the correlation of adjacent pixels between the original image and the image after re-encryption by the POB system. First, 6000 adjacent pixel pairs are randomly selected from the original image and the re-encrypted image in the horizontal, vertical, and diagonal directions; then, the correlation of these adjacent pixels is calculated, and simulation tests are carried out. Figure 9 shows the correlation of adjacent pixels between the four original ROS images and the re-encrypted images. It is obvious from the figure that the correlation coefficient of the original image is close to 1, indicating that there is a strong correlation between adjacent pixels, while after re-encryption, there is a lower correlation between adjacent pixels in the image.



(**a**)

Figure 9. *Cont.*



(**b**)

(**c**)

(**d**)

Figure 9. The correlation comparison of the original and re-encrypted image. (**a**) Brain 01; (**b**) Brain 02; (**c**) Pulmonary 01; (**d**) Pulmonary 02.

4.2. Hiding Performance

4.2.1. Embedding Capacity Analysis

This part will analyze the embedding capabilities of the scheme in this paper. To intuitively measure the embedding capacity, it is usually expressed by the embedding rate (ER), which is the amount of data that can be embedded in each pixel on average, calculated by Equation (8):

$$ER = \frac{n(bit)}{M \times N}(bpp) \tag{8}$$

where n is the bits of the embedded data and $M \times N$ is the number of pixels in the embedded image.

Because the scheme proposed in this paper is based on the POB algorithm for data hiding, two extra pieces of data can be attached to each pixel, so the embedding capacity is very large. In addition, the method of separation of shares is adopted in this paper, and data embedding can be carried out in both ShareA″ and ShareB″. To protect the sensitive areas of medical images, hiding can be given priority in ShareB″. If the secret data are embedded in a share of a 256 × 256 image, the embedding rate can reach 2, and the maximum embedding capacity is 2 × 65,536 bits. Compared with the previous methods of using the POB system to hide data in images, the embedding capacity of this paper's method is greatly improved.

Table 1 shows the maximum embedding capacity and maximum embedding rate of our scheme and the scheme of [19] in four medical images. The embedding in the scheme of [19] is only carried out at the peak and sub-peak points of the image, whereas our scheme has a very high embedding capacity because it can embed secret messages in all pixels of the image.

Table 1. Comparison of maximum embedding capacity and maximum embedding rate between our scheme and the scheme of [19] in four test images.

Test Images	ROS	Our Scheme (Encrypted ShareA')		Scheme of [19]	
		Maximum Embedding Capacity (bits)	Maximum Embedding Rate (bpp)	Maximum Embedding Capacity (bits)	Maximum Embedding Rate (bpp)
Brain 01(512 × 512)	352 × 352	2 × 262,144	2	263,216	1.0041
Brain 02 (512 × 512)	464 × 464	2 × 262,144	2	185,322	0.7069
Pulmonary 01 (256 × 256)	176 × 176	2 × 65,536	2	93,598	1.4282
Pulmonary 02 (256 × 256)	176 × 176	2 × 65,536	2	94,742	1.4456
Pancreas (180 × 180)	120 × 120	2 × 32,400	2	5808	0.1793
Chest (256 × 256)	228 × 188	2 × 65,536	2	47,166	0.7197
Knee (300 × 162)	212 × 162	2 × 48,600	2	29,028	0.5973
Hand (64 × 64)	46 × 60	2 × 4096	2	2058	0.5024

In addition, the ER of our scheme in common images is compared with that of several related information-hiding methods. Figure 10 shows that the ER of our scheme is significantly higher than that of other schemes. Meanwhile, the ER of our scheme is a constant value that is not affected by the image distribution. Among these existing methods, the embedding rate of some methods depends on the image distribution. Images with a smooth texture can obtain a higher embedding rate, while those with a complex texture have a lower embedding rate, so the proposed method can avoid this problem.

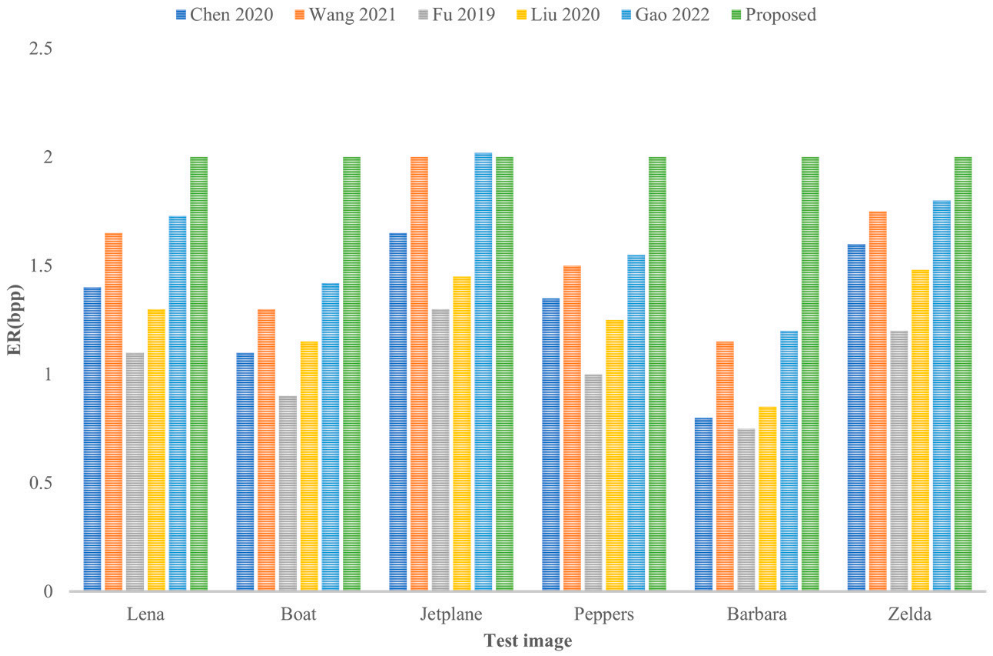


Figure 10. Comparison of ER between our scheme and related schemes [5–9] in ordinary images.

4.2.2. PSNR and SSIM Analysis

The PSNR (peak signal-to-noise ratio) is an important index to measure image quality. It is often used to compare the difference between two images or to measure the effect of an algorithm on an image. The PSNR is calculated as follows:

$$PSNR = 10 \times lg(\frac{I^2{}_{max}}{MSE}) \tag{9}$$

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(P(i,j) - C(i,j))^2 \tag{10}$$

where $M \times N$ represents the image's size, $C(i,j)$ is the recovered image's gray value of the pixel, $P(i,j)$ is the original image's gray value of the pixel, and I_{max} is the maximum pixel value of the original image.

SSIMs (structural similarity index metrics) are often used to assess the degree of similarity between two images. SSIMs can be calculated by Equation (11):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{11}$$

where x and y denote the two images, μ_x and μ_y denote their means, σ_x^2 and σ_y^2 represent their variances, σ_{xy} represents their covariances, and C_1 and C_2 are two constants used to ensure calculation stability. The value range of the SSIM is [−1, 1]. When two images are identical, the SSIM is 1.

Table 2 shows that the PSNR and SSIM values of the restored image are obtained after data of different sizes are embedded into four images of different sizes through the proposed scheme. The experimental results show that the SSIM value of the original image and the restored image in the proposed scheme is 1, further demonstrating the reversibility of the scheme.

Table 2. PSNR and SSIM of recovered images under different embedding capacities.

Test Images		Embedding Method	Embedded Capacity (bits)	PSNR	SSIM
512 × 512	Brain 01	Encrypted ShareA′	52,428 (10%)	∞	1
512 × 512	Brain 02	Encrypted ShareB′	157,286 (30%)	∞	1
256 × 256	Pulmonary 01	Encrypted ShareA′	65,536 (50%)	∞	1
256 × 256	Pulmonary 02	Encrypted ShareB′	104,857 (80%)	∞	1
180 × 180	Pancreas	Encrypted ShareA′	12,960 (20%)	∞	1
256 × 256	Chest	Encrypted ShareB′	19,660 (15%)	∞	1
300 × 162	Knee	Encrypted ShareB′	29,160 (30%)	∞	1
64 × 64	Hand	Encrypted ShareA′	3276 (40%)	∞	1

4.2.3. Time Efficiency Analysis

The experiments are implemented on a computer with a 1.60 GHz Intel i5 processor, 8.00GB memory, and Windows 10 operating system. Using the POB algorithm, because of the simultaneous compression and re-encryption operations, will lead to a slightly higher time overhead in programming implementation. Table 3 lists the time overhead of using single POB and double POB embedding in different embedding capacities. In the table, it can be seen that although the use of double POB embedding has a slightly larger time overhead, it does not affect the practical application.

Table 3. Comparison of time efficiency between single POB and double POB embedding in the same image with different embedding capacities.

Test Images		Embedding Method	Embedded Capacity (bits)	Single POB (ms)	Double POB (ms)
512 × 512	Brain 01	Encrypted ShareA′	52,428 (10%)	9740.6	21,453.5
	Brain 01	Encrypted ShareA′	157,286 (30%)	11,404.6	24,227.8
256 × 256	Pulmonary 01	Encrypted ShareA′	65,536 (50%)	4594.2	8070.3
	Pulmonary 01	Encrypted ShareA′	104,857 (80%)	4752.9	8586.2

4.3. Recovery Evaluation

In order to verify the performance of the tamper recovery scheme in the proposed scheme, ShareA″ of Brain 01 is used as an example here, and an attack test is carried out on ShareA″ with embedded secret data and authentication bits. Three classic attack methods are adopted, including content cropping, text addition, and content exchange. The attack results and tampering recovery results are shown in Figure 11.

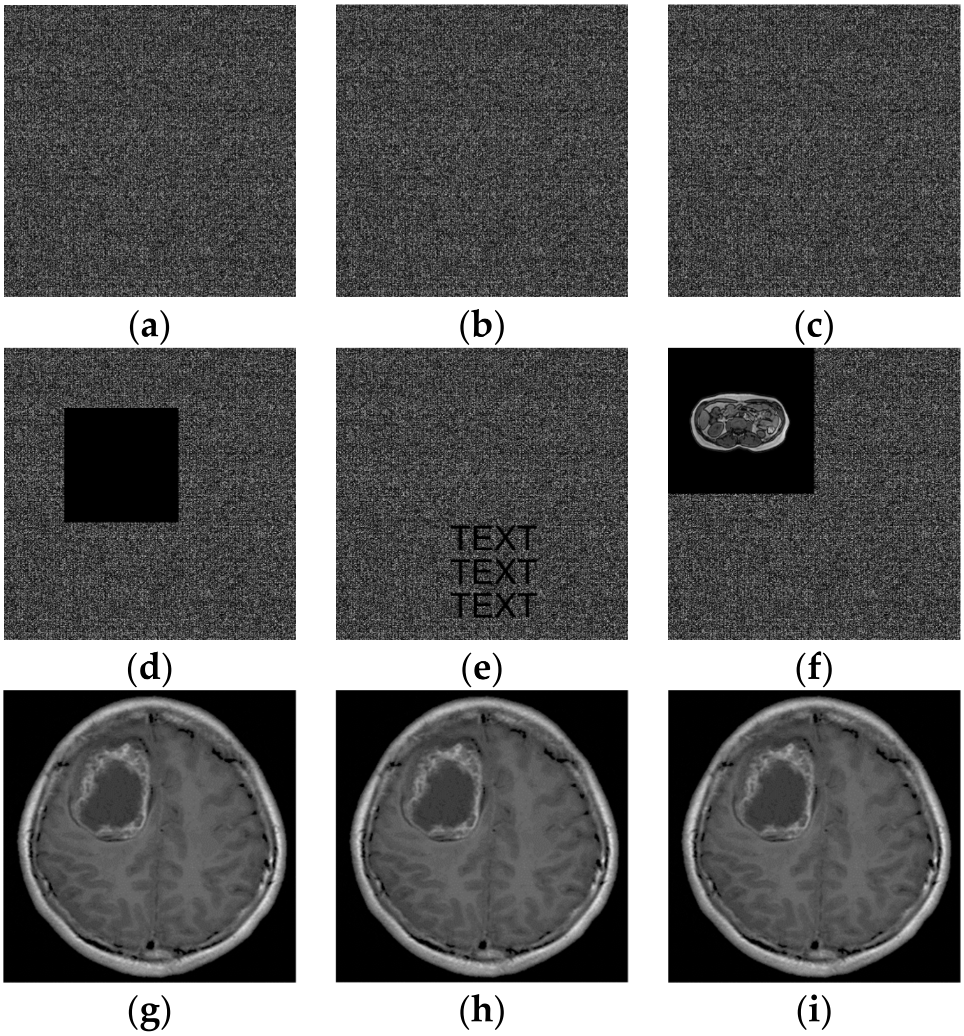


Figure 11. Results of attack test. (**a**–**c**) ShareA″; (**d**–**f**) the attacks of content cropping, text addition, and content exchange carried out on ShareA″; (**g**–**i**) the recovered ShareA.

The first column shows a content cropping attack on ShareA″, from which a block with a size of 200 × 200 is cropped. The second column performs a text addition attack on ShareA″. The third column is replaced with "Pulmonary 01" in the upper left corner of

ShareA″; its size is 256 × 256, and the last line represents the recovery result of ShareA″ after the attack. In addition, the same attack was tested on ShareB″, and the same results were obtained.

Four different images were tested for different tampering methods and recovered; Table 4 shows the PSNR and SSIM values of the recovered images.

Table 4. PSNR and SSIM of the recovered image under different tampering methods.

Test Images		Tampered Region	Tampering Methods	PSNR	SSIM
512 × 512	Brain 01	ShareA″	cropping (200 × 200)	∞	1
	Brain 02	ShareB″	exchange (256 × 256)	∞	1
256 × 256	Pulmonary 01	ShareA″	text addition	∞	1
	Pulmonary 02	ShareB″	cropping (50 × 50)	∞	1

ShareA″ShareB″ShareA′′ShareB″

4.4. Comparison with the Related Schemes

The comparison results between the proposed scheme and some related schemes are shown in Table 5.

Table 5. The comparison between the proposed scheme and some related schemes.

Scheme	Watermarking Consideration	Reversibility	ROI Consideration	Share Consideration	Tamper Recovery Consideration
Priyanka [1]	Robust	No	Yes	No	No
Wang [2]	Fragile	No	Yes	No	No
Thakur [3]	Robust	No	No	No	No
Shehab [15]	Fragile	No	No	No	No
Parah [16]	Fragile	Yes	No	No	No
Gao [17]	Fragile	Yes	Yes	No	No
Li [23]	None	No	No	Yes	Yes
Singh [20]	Robust	Yes	/	Yes	No
Ren [19]	Robust	Yes	/	No	No
Ours	Robust	Yes	Yes	Yes	Yes

The algorithm in [3] is designed to irreversibly embed message bits to resist attacks, but it cannot recover the original medical image after extracting information bits. For authentication and integrity reasons, the algorithms in [2,15] embed message bits in a vulnerable way, but they still cannot recover the original medical image. Compared with other algorithms, the algorithm in [16] modifies the sensitive pixel values in the ROI to a greater extent, which is not allowed in the use of medical images. Although the algorithm in [17] considers the ROI, the difference between the watermark image and the original image is still large. Although the algorithm in [23] adopts the secret sharing method to detect tampering, it does not have the function of embedding watermark messages, and it does not consider the sensitive pixels in the ROI. Although POB system based on secret sharing was considered in [20], it could not guarantee lossless recovery. In [19], a method based on POB system was used to hide data in the peak point, but it did not have the function of preventing tamper detection. The method in this paper not only carries additional secret data during image transmission but also makes use of secret sharing to increase the embedding capacity of data and protect the content security of medical images. In addition, it also has an authentication function that can recover the original image through lossless recovery, which is very suitable for the needs of medical images, and it has better performance in data hiding and lossless recovery.

4.5. Application Scenario of the Proposed Scheme

In the application of telemedicine, there will be attacks to tamper with, steal, and forge patient private information or medical image content, and these attacks may lead to a wrong diagnosis by the receiver and produce serious medical accidents, which is firmly not allowed. Therefore, in view of these possible attacks, while the secret information is embedded, the scheme in this paper also carries out an identity authentication on the medical image, as shown in Figure 12, which can not only protect the personal information of patients but also ensure the integrity and security of the medical image.

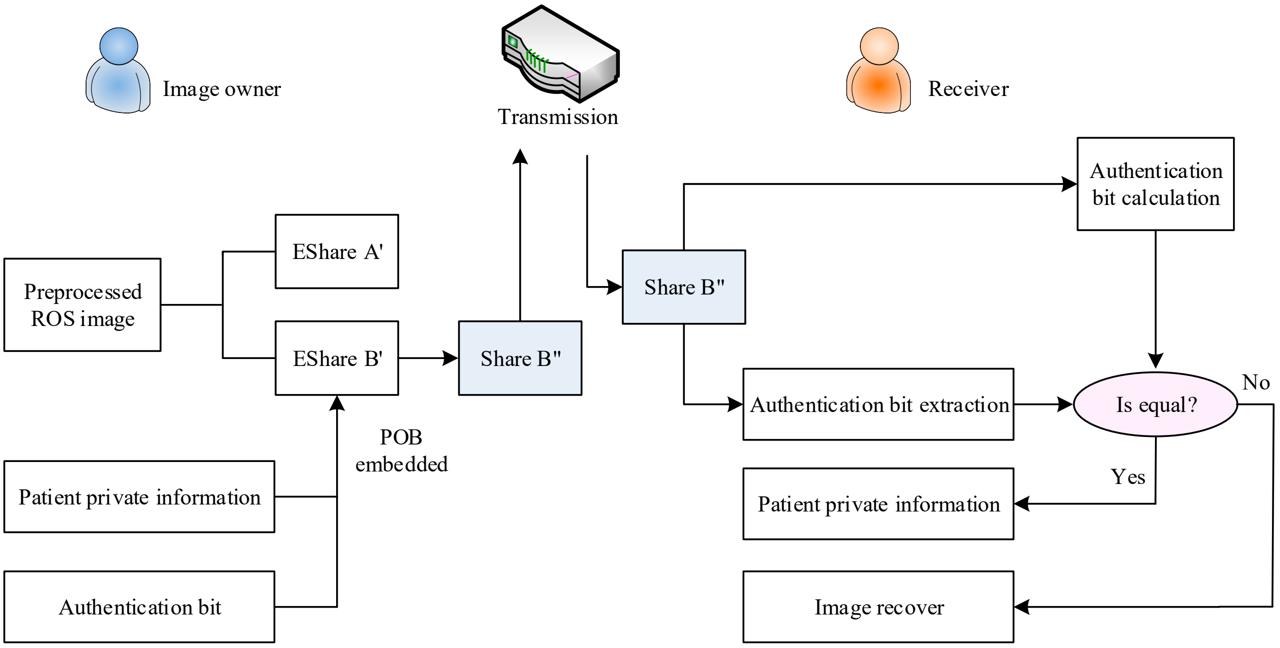


Figure 12. The application scenario of the proposed scheme.

5. Conclusions

This paper proposes a new medical image hiding and authentication algorithm based on double POB system. Firstly, the ROS is extracted, and then the ROI and RONI are separated by an image segmentation algorithm. After bit plane separation and cross-reorganization, two encrypted shares are obtained through the lossless compression and image encryption algorithm. In the double-hiding process, the secret message and authentication bits are embedded using the POB number system. In the recovery process, authentication is carried out first; if it is not tampered with, the receiver will extract the secret message and complete the recovery of the original image. If the authentication fails, the repeatedly filled data are directly used for lossless recovery.

The experimental results showed that the scheme in this paper significantly enhances the embedding capacity without damaging the perceptual quality of the image. Compared with other schemes, it has good performance in data embedding and lossless recovery. Additionally, the bit plane separation and cross-reorganization method effectively safeguard highly sensitive pixels within the ROI of medical images, which is crucial for their utilization. However, when using double POB system, compression and re-encryption operations are performed at the same time, which leads to a slightly higher time overhead. In future work, we will mainly focus on how to reduce the time cost of using the double POB function.

Author Contributions: Conceptualization, X.S.; Methodology, X.S.; Software, X.S. and E.T.; Writing—review and editing, F.R.; Formal analysis, E.T.; Investigation, M.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the National Nature Science Foundation of China (Program No. 62202377), the Natural Science Basic Research Plan of Shaanxi Province of China (Program No. 2024JC-YBMS-524, 2022JM-353), the Key Research and Development Program of Shaanxi (Program No. 2023-YBGY-015), and the Scientific Research Program Funded by Shaanxi Provincial Education Department (No. 22JK0560).

Institutional Review Board Statement: Not applicable for studies not involving humans or animals.

Informed Consent Statement: Not applicable for studies not involving humans.

Data Availability Statement: All data generated or analyzed during this study are included in this published article. The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Priyanka; Maheshkar, S. Region-based hybrid medical image watermarking for secure telemedicine applications. *Multimed. Tools Appl.* **2017**, *76*, 3617–3647. [CrossRef]
2. Wang, D.; Chen, D.; Ma, B.; Xu, L.; Zhang, J. A high capacity spatial domain data hiding scheme for medical images. *J. Signal Process. Syst.* **2017**, *87*, 215–227. [CrossRef]
3. Thakur, S.; Singh, A.K.; Ghrera, S.P.; Elhoseny, M. Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications. *Multimed. Tools Appl.* **2019**, *78*, 3457–3470. [CrossRef]
4. Lee, H.Y. Adaptive reversible watermarking for authentication and privacy protection of medical records. *Multimed. Tools Appl.* **2019**, *78*, 19663–19680. [CrossRef]
5. Chen, K.M. High capacity reversible data hiding based on the compression of pixel differences. *Mathematics* **2020**, *8*, 1435. [CrossRef]
6. Wang, Y.; He, W. High capacity reversible data hiding in encrypted image based on adaptive MSB prediction. *IEEE Trans. Multimed.* **2021**, *24*, 1288–1298. [CrossRef]
7. Fu, Y.; Kong, P.; Yao, H.; Tang, Z.; Qin, C. Effective reversible data hiding in encrypted image with adaptive encoding strategy. *Inf. Sci.* **2019**, *494*, 21–36. [CrossRef]
8. Liu, Z.L.; Pun, C.M. Reversible data hiding in encrypted images using chunk encryption and redundancy matrix representation. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1382–1394. [CrossRef]
9. Gao, K.; Horng, J.H.; Chang, C.C. High-capacity reversible data hiding in encrypted images based on adaptive block encoding. *J. Vis. Commun. Image Represent.* **2022**, *84*, 103481. [CrossRef]
10. Qin, C.; Zhang, W.; Cao, F.; Zhang, X.; Chang, C.C. Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Process.* **2018**, *153*, 109–122. [CrossRef]
11. Singh, D.; Singh, S.K. DCT based efficient fragile watermarking scheme for image authentication and restoration. *Multimed. Tools Appl.* **2017**, *76*, 953–977. [CrossRef]
12. Singh, D.; Singh, S.K. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image Represent.* **2016**, *38*, 775–789. [CrossRef]
13. Tiwari, A.; Sharma, M.; Tamrakar, R.K. Watermarking based image authentication and tamper detection algorithm using vector quantization approach. *AEU-Int. J. Electron. Commun.* **2017**, *78*, 114–123. [CrossRef]
14. Gull, S.; Loan, N.A.; Parah, S.A.; Sheikh, J.A.; Bhat, G.M. An efficient watermarking technique for tamper detection and localization of medical images. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1799–1808. [CrossRef]
15. Shehab, A.; Elhoseny, M.; Muhammad, K.; Sangaiah, A.K.; Yang, P.; Huang, H.; Hou, G. Secure and robust fragile watermarking scheme for medical images. *IEEE Access* **2018**, *6*, 10269–10278. [CrossRef]
16. Parah, S.A.; Ahad, F.; Sheikh, J.A.; Bhat, G.M. Hiding clinical information in medical images: A new high capacity and reversible data hiding technique. *J. Biomed. Inform.* **2017**, *66*, 214–230. [CrossRef] [PubMed]
17. Gao, G.; Wan, X.; Yao, S.; Cui, Z.; Zhou, C.; Sun, X. Reversible data hiding with contrast enhancement and tamper localization for medical images. *Inf. Sci.* **2017**, *385*, 250–265. [CrossRef]
18. Sreekumar, A.; Sundar, S.B. An efficient secret sharing scheme for n out of n scheme using POB-number system. *Hack* **2009**, *1*, 33–37.
19. Ren, H.; Niu, S.; Wang, X. Reversible data hiding in encrypted images using POB number system. *IEEE Access* **2019**, *7*, 149527–149541. [CrossRef]

20. Singh, P.; Raman, B.; Agarwal, N. Toward encrypted video tampering detection and localization based on POB number system over cloud. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2116–2130. [CrossRef]
21. Xiang, Y.; Xiao, D.; Wang, H.; Li, X. A secure image tampering detection and self-recovery scheme using POB number system over cloud. *Signal Process.* **2019**, *162*, 282–295. [CrossRef]
22. Liu, Y.; You, Z.; Gao, T. Lossless image hierarchical recovery based on POB number system. *Signal Process.* **2020**, *167*, 107293. [CrossRef]
23. Li, Q.; Fu, Y.; Zhang, Z.; Fofanah, A.J.; Gao, T. Medical images lossless recovery based on POB number system and image compression. *Multimed. Tools Appl.* **2022**, *81*, 11415–11440. [CrossRef] [PubMed]
24. Wang, X.; Wang, M. A hyperchaos generated from Lorenz system. *Phys. A Stat. Mech. Its Appl.* **2008**, *387*, 3751–3758. [CrossRef]
25. Erdal, E. Huffman-based lossless image encoding scheme. *J. Electron. Imaging* **2021**, *30*, 053004. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.