# Reversible information hiding algorithm based on difference histogram displacement

**Pengcheng Zhu[a,*], Chenge Luo[b]**

School of Information Engineering, Wuhan Huaxia Institute of Technology, Wuhan, Hubei Province, China

[a]zpctips@163.com, [b]3125791079@qq.com

*Corresponding author

**Keywords:** reversible watermarking technology, lossless extraction, carrier image restoration, data embedding, prediction error threshold, statistical properties, difference domain histogram modification, gray value sorting, watermark embedding, peak points, zero points, histogram translation method, separable reversible information hiding algorithm, ciphertext domain, homomorphicencryption

**Abstract:** The paper presents a significant homomorphic encryption scheme for signal processing within the ciphertext domain, allowing for direct operations on encrypted data. The methodology involves a pixel grouping strategy and block difference histogram technique, which not only increase the embedding capacity but also maintain image quality standards. During the information embedding phase, a histogram shifting approach is employed, leveraging edge values to embed without the need for auxiliary information, thereby enhancing security.The embedding process comprises single-layer embedding with left or right shifts based on the secret information. A critical aspect addressed in this scheme is the overflow problem, which is managed through adjustments in grayscale values and the recording of overflow instances. By implementing these strategies, the algorithm enhances the embedding capacity without compromising security, making it suitable for reversible information hiding applications in the realm of sensitive imaging.Overall,this homomorphic encryption scheme presents a comprehensive approach to secure data operations in the ciphertext domain, offering a balance between embedding capacity and image quality preservation, thereby proving to be a valuable contribution to the field of reversible information hiding and encryption technologies.

## 1. Introduction

In the field of sensitive imaging, reversible watermarking technology has attracted much attention and is widely used because of its ability to achieve lossless extraction of watermarks and restore the carrier image. One of the earliest reversible data embedding efforts was a patent filed by Barton in 1994. In recent years, reversible information hiding methods are mainly divided into three categories: based on lossless compression, based on difference expansion, and based on gray value modification[1].

Some other literature proposes a multi-level reversible watermarking algorithm based on difference domain histogram modification, which improves the watermark embedding capacity by calculating the difference and selecting appropriate embedding points. However, some of these methods may select edges or noise points of the image as reference pixels, resulting in less dense calculated differences.

This paper proposes a separable reversible information hiding algorithm in the ciphertext domain. This algorithm is based on the histogram translation algorithm. It builds a difference histogram through image block and detects edge values to improve the embedding capacity, making the image recovery and information extraction process completely reversible. At the same time, homomorphic encryption is used to realize the separability of the image decryption process[2].

  DOI: 10.25236/iwmecs.2024.016

## 2. Manuscript Preparation

### 2.1. Page Setup

Homomorphic Encryption Scheme Homomorphic encryption technology is of great significance to the development of signal processing technology in the ciphertext domain. Its characteristic is that it allows direct operations on ciphertext data in the encryption domain. The description of the homomorphic encryption process in the literature can be expressed as (1) means.

$$E(m_1 \odot Mm_2) = c_1 \odot Cc_2 \tag{1}$$

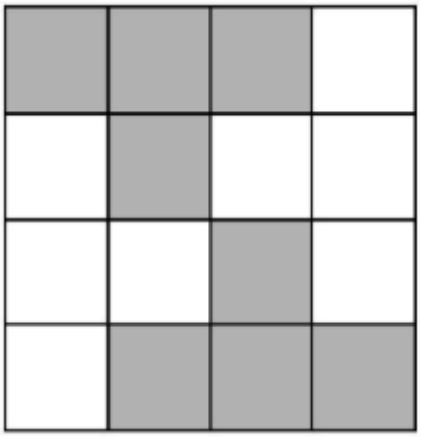


Figure 1 Pixel grouping scheme

We first use the $4 \times 4$ pixel block as the basic unit and divide it into 4 non-overlapping groups of pixels(Fig.1). Each group is composed of 4 pixels with a "T"-shaped positional relationship. The same group of 4 pixels share an encryption key $k$. Assume that the number of pixels in the image is $n$, and then the length of the pseudo-random number sequence generated by the $RC4$ algorithm is $1/4n$.Addition operation, as shown in formula (2).

$$\begin{aligned} C &= E(M', K) = (M' + K) \bmod 256 \\ c_{i,j} &= (m_{ij} + k_{ij}) \bmod 256 \end{aligned} \tag{2}$$

Among them, $i$, $j$ represent the two-dimensional coordinates of the image, $M'$ is the preprocessed image, $C$ is the encrypted image, $K$ is a pseudo-random number sequence, $k_{ij}$ represents the $K$ value corresponding to the pixel with coordinates of $i$, $j$ and the pseudo-random number $K$ sequence can be used as the encryption of the image key.

### 2.2. Block difference histogram

First, the image is divided into blocks of size s*s, and must satisfy:

$$s \bmod 4 = 0, 4 \le s \le \min(i_{\max}, j_{\max}) \tag{3}$$

After the block is completed, each block of the image is scanned separately. According to the grouping method in the homomorphic encryption process, each block is grouped. Each "T" shaped pixel group can obtain 3 difference values, thereby obtaining the difference of each block of image. value histogram [3]. The algorithm in this article is a direct grouping method. On the basis of ensuring that pixels are only modified once at most, the number of differences obtained is higher than that of similar algorithms, reaching $3/4n$. The “T” shaped pixel group is shown in Figure 2.

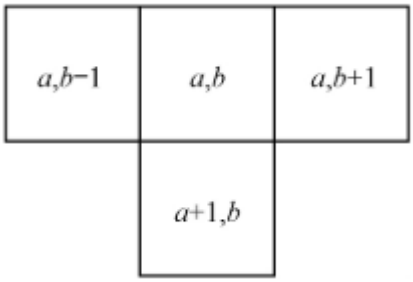



Figure 2 “T” shaped pixel group

Taking the pixel group shown in position relationship diagram 2 as an example, the following formula can be obtained:

$$\begin{aligned} d_{a,b-1} &= (c_{a,b-1} - c_{a,b}) \bmod 256 \\ d_{a+1,b} &= (c_{a+1,b} - c_{a,b}) \bmod 256 \end{aligned} \tag{4}$$

It can be proved by equation (4) that performing a difference on the ciphertext image is equivalent to directly operating on the plaintext image.

### 2.3. Information embedding

#### 2.3.1. Histogram shifting scheme

Figure 3 is the complete difference histogram of the encrypted Lena image obtained by the above method. It can be seen that the number of differences near 0 and 255 is the largest, indicating that the majority of adjacent pixel differences are small, which is consistent with the continuous nature of natural image pixels. In the embedding process, the peak value of the difference histogram is generally selected as the embedding position. Therefore, the algorithm in this paper uses the edge values of the block difference histogram to embed to improve the embedding capacity, and does not require auxiliary information. Information embedding is achieved by shifting the difference histogram.[4] Taking the $8\times8$ size block as an example, the left edge value is marked as $L_1, L_2, L_3, L_4 ...$

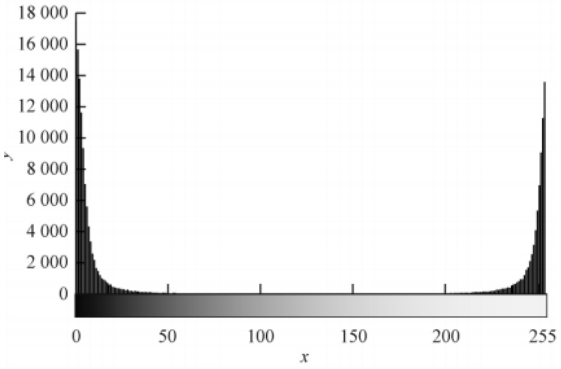


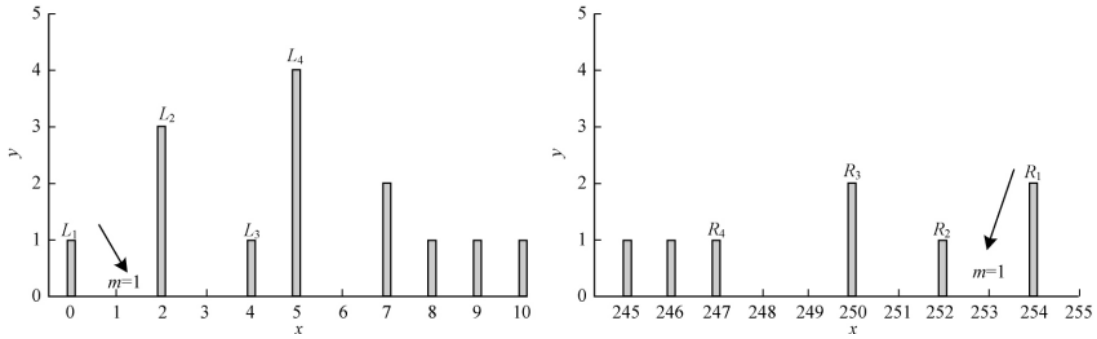

Figure 3 Difference histogram of the complete Lena diagram

#### 2.3.2. Embedding steps

The embedding steps are as follows:

1) Single layer embedding

After the difference histogram translation is completed, the right side of L1 and the left side of R1 have given space for secret information to be embedded in the first layer. L1 and R1 are the embedding positions, and the number of L1 and R1 is the embedding capacity. Single layer The embedding process is shown in Figure 4.




(a) Embedding on the left (b) Embedding on the right

Figure 4 Single layer embedding process

When the hidden information $m=0$, $L_1$ and $R_1$ do not change. When $m=1$, $L_1$ is translated one bit to the right or $R_1$ is translated one bit to the left. After one image is embedded, the same operation is performed on the next image. The above operations can be summarized by the following formula:

$$c_{i,j}=\begin{cases} c_{i,j+1}, m=1, d_{i,j}=L_1 \\ c_{i,j}, m=0 \\ c_{i,j}, m=1, d_{i,j}=R_1 \end{cases}$$

If $d_{a,b-1}, d_{a,b+1}, d_{a+1,b}$ moves in the same direction, as shown in equation (5).

$$c_{a,b}=\begin{cases} c_{a,b-1}, m=1, d_{i,j}=L_1 \\ c_{a,b+1}, m=1, d_{i,j}=R_1 \end{cases} \tag{5}$$

As can be seen from Figure 4(a), there is a case where the difference value of the block difference histogram is 0, $L_1=0$, that is to say, after all image blocks complete information embedding, all positions with a difference value of 0 have secret information embedded, The embedding position is the same as the full difference histogram. However, some difference histograms may not have a difference value of 0. In this case, in addition to embedding the position where the difference value is 0, the block embedding also embeds secret information in other positions where the difference value is greater than 0, such as In Figure 4(b), $R_1=254$ ,unlike the embedding position of the complete difference histogram, information is also embedded at the position of difference 254, which increases the embedding capacity compared to the original peak embedding.[5]

### 2.3.3. Overflow problem

The difference histogram translation and information embedding are implemented through $c_{i,j+1}$ or $c_{i,j-1}$. In an 8-bit grayscale image, the value range of $c_{i,j}$ is 0 to 255, which means that the sum of $c_{i,j+1}$ and $c_{i,j-1}$ may change the pixel value. If it is -1 or 256, overflow will occur. The original image $M$ needs to be preprocessed, modify the points with pixel values 0 or 255, and find the grayscale with the smallest number of pixels between [0, 127] and [128, 255]. Values, recorded as *left* and *right*, all pixels with grayscale values between [0, *left* +1] are increased by 1, and all pixels with grayscale values between [ *right* +1, 255] are reduced by 1, to obtain $M'$. In other cases, it is recorded as 1. is 0. This information needs to be combined with the length of the secret information to form a one-dimensional array *head*, placed at the beginning of the secret information that needs to be embedded, and embedded together with the secret information at each layer of embedding.

### 2.4. Information extraction and image decryption

#### 2.4.1. Extract and restore

Divide the confidential image B into image blocks of size $s \times s$ according to the value of $s$ in the hidden key, and then extract from the highest layer according to the number of embedding layers provided by the key, taking 4 layers of embedding as an example:

1) Perform "T"-shaped grouping according to Figure 2 to obtain the block difference histogram. Determine the embedding position L4 based on the block difference histogram. R4 then scans the difference values of the image blocks in order, $d_{ij} = L_4$ or $R_4$ recorded as 0, $d_{ij} = L_4 + 1$ or $R_4 - 1$ recorded as 1, put them in the one-dimensional array $N_4$ in order.

2) Restore the image, offset all modifications made during the translation of the difference histogram and the embedding process, then: the pixels need to be modified according to the one-dimensional array $map$ recorded in $N_4$ and the length of the $map$.

3) Extract and restore the next image block, repeat steps 1) and 2) until all image blocks are processed.

4) Repeat steps 1) to 3) to extract the next layer to obtain the encrypted image C without secret information and the information embedded in the first three layers $N_1, N_2, N_3$. Integrate them in order from the bottom to the top to obtain the complete secret. Information $N$.

#### 2.4.2. Image decryption

After extraction and restoration, the image can be directly decrypted to obtain the preprocessed original image. The same random number sequence K used in image encryption can be constructed through the encryption key seed. Then it can be decrypted according to the following formula:

$$\begin{aligned} M' &= D(C,K) = (C-K) \bmod 256 \\ (c_{i,j} - k_{i,j}) \bmod 256 &= m_{i,j} \end{aligned} \tag{6}$$

Among them, $D()$ represents the decryption operation, and $M'$ is the original image after preprocessing. The lossless original image $M$ can be obtained by performing the reverse preprocessing operation based on the information provided by $pre$.

You can also directly decrypt the image to obtain a plaintext image containing secret information $B$. It can be seen that the information extraction and image decryption processes of this algorithm are separable[6].

### 3. Experimental results and analysis

General indicators to evaluate the performance of information hiding algorithms include imperceptibility, embedding capacity and error rate. Imperceptibility requires that the change to the carrier is not easily detectable [7]. Subjectively, it can be perceived through vision, and objectively, the peak signal-to-noise ratio ($PSNR$) can be used to evaluate the degree of distortion of the carrier. The calculation formula is:

$$P_{PSNR} = 10 \lg \frac{255^2}{M_{MSE}} \tag{7}$$

Among them, $MSE$ is the mean square error between the image after embedding information and the original image, and its calculation formula is:

$$M_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (c_i - m_i)^2 \tag{8}$$

In order to clarify the various indicators of the algorithm, this article uses Matlab to conduct simulation experiments. In order not to lose generality, the embedded information is a sequence of 0

and 1 randomly generated by the rand() function. The experimental test images are all large and small grayscale levels of 8. Standard image, as shown in Figure 5.


Figure 5 Experimental test image

### 3.1 Result of research

The embedding capacity through multi-layer embedding algorithms will continue to increase. Generally speaking, the higher the embedded rate, the better. However, as the number of embedding layers increases, more and more pixels are modified, which will inevitably reduce the imperceptibility. The two constrain each other. The entropy value of the image is maximized after encryption, and it is meaningless to discuss the *PSNR* of the encrypted image. The following is a discussion of the image after direct decryption. Table 1 shows the changes in *PSNR* of the test image during the four-layer embedding process. The higher the value, the better the image quality.

Table 1 PSNR of Directly Decrypted Images

| Test image | First layer | Second layer | Third layer | Fourth layer |
|---|---|---|---|---|
| Lena | 50.81 | 45.66 | 43.00 | 41.60 |
| Plane | 50.97 | 46.00 | 43.70 | 42.44 |
| Peppers | 48.93 | 45.25 | 42.89 | 41.33 |
| Goldhill | 50.75 | 45.41 | 42.65 | 40.82 |
| Baboon | 50.58 | 45.16 | 42.12 | 40.02 |
| Barbara | 50.79 | 45.46 | 42.65 | 40.98 |
| Lake | 51.01 | 45.52 | 42.79 | 40.99 |
| Boat | 50.85 | 45.49 | 42.60 | 40.80 |

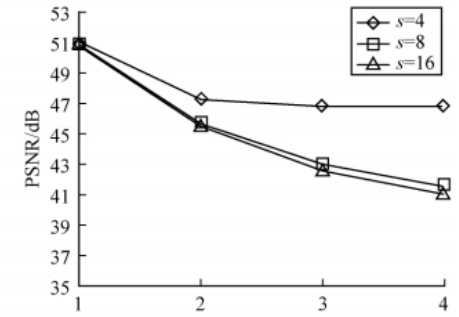
### 3.2 Image tiling

The image is divided into blocks and then the information is embedded. Compared with embedding the entire image directly, information is also embedded in the original non-peak areas of the difference histogram, which can increase the embedding capacity. It can be seen from Figure 7(a) that when $s = 4$, the embedding capacity of the single layer is significantly higher than other cases, and the *PSNR* value of the image remains at a high level, but it can be seen from Figure 7(b) , as the number of embedding layers increases, the growth rate of the embedding rate decreases sharply. In this case, only the edge values are embedded. Most image blocks do not meet the embedding conditions in the third layer. This is the direct reason restricting capacity growth.

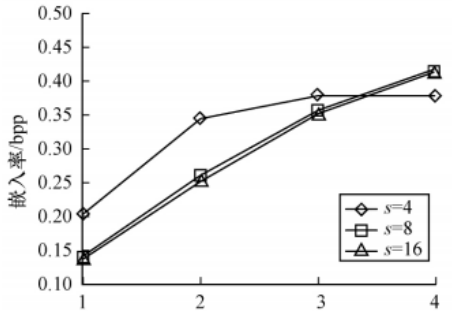
## 4. Conclusion

This paper proposes a reversible information hiding algorithm in the ciphertext domain based on difference histogram translation. This algorithm selects the difference histogram translation as the entry point, uses the grouping method to obtain more differences, and breaks through the limit of the embedding capacity by the difference histogram peak through image blocking and edge difference embedding, thereby improving the embedding rate. Experimental results show that the

algorithm can achieve completely reversible information hiding in the ciphertext domain, and can operate flexibly when performing information extraction and image decryption at the receiving end. This article only analyzes grayscale images, and research on information hiding in color images will be the next research direction.




(a) PSNR of different blocks




(b) Embedding rate of different blocks

Figure 6 Experimental results of different blocks

See figure 6 ,when the amount of embedded information is small, dividing the image into $4\times4$ blocks can maximize the value *PSNR* and optimize the algorithm performance. Comparing $s = 8$ and $s = 16$ block schemes, both *PSNR* and *MSE* have little difference in embedding at each layer, and when $s = 8$ and *MSE* are both slightly higher than $s = 16$, that is, when $s > 8$ Afterwards, the algorithm performance starts to decrease. If the amount of embedded information is large and the maximum capacity of $s=4$ cannot be required, $s=8$ should be selected to block the image to obtain the best performance while ensuring the embedding capacity.

## References

[1] Zhou S Y, Zhang W M, Shen C M. Rate-distortion model for grayscale-invariance reversible data hiding[J]. Signal Processing, 2020, 172: 107562

[2] Jhong C L, Wu H L. Exploring more capacity for grayscale-invariance reversible data hiding[J]. IEEE Access, 2021, 9: 137005-137014

[3] Jhong C L, Wu H L. Grayscale-invariant reversible data hiding based on multiple histograms modification[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(9): 5888-5901

[4] Bhatnagar P, Tomar P, Naagar R, et al. Reversible Data Hiding scheme for color images based on skewed histograms and cross-channel correlation[C]//2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT). Bhubaneswar, India. IEEE, 2023: 419-426

[5] Chang Q, Li X L, Zhao Y. Reversible data hiding for color images based on adaptive three-dimensional histogram modification[J]. IEEE Transactions on Circuits and Systems for Video

Technology, 2022, 32(9): 5725-5735

[6] Sachnev V, Kim H J, Nam J, et al. Reversible watermarking algorithm using sorting and prediction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(7): 989-999

[7] Gui X L, Li X L, Yang B. A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding[J]. Signal Processing, 2014, 98: 370-380