Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 4
ICCIA 2024

# RAINBOW: Resilient Asymmetric Imaging Non-linear Bit-level Ordering with Hyperchaotic Operation for Color image Encryption

Wenhui Zhang

College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

Zwenhui@hnu.edu.cn

**Abstract.** Hyperchaotic encryption, known for its high level of unpredictability and complexity, is widely used in the field of image encryption. However, current hyperchaotic image encryption techniques have certain limitations, particularly in terms of their simplistic processing and lack of depth in layer interaction. These limitations ultimately hinder their effectiveness in ensuring security. In order to overcome these challenges, we propose RAINBOW, a method that integrates bit-level and pixel-level permutation and diffusion across color layer planes. RAINBOW accomplishes a highly complex and diversified permutation process by leveraging the division of bit planes and pixel-level manipulation across color layers. Moreover, we utilize a cross-layer three-dimensional approach in the diffusion process. This approach ensures that the current pixel is influenced by around adjacent pixels, effectively strengthening the image's security. Through extensive testing on various public color im- age datasets, our results demonstrate that the RAINBOW scheme significantly improves encryption quality and effectively mitigates multiple types of attacks.

**Keywords:** Cross-plane, Hyperchaotic, Color Image Encryption.

## 1. Introduction

With the rapid advancement of technology, network communication has evolved from text-based to multimedia, notably images [1]. Images, being rich in information and intuitive, find extensive use in education [2], medical diagnostics [3], and military applications [4], often containing sensitive data. However, transmitting images over insecure channels poses privacy risks [5], necessitating secure image transmission methods. This has led to a surge in research on image encryption techniques [6][7].

To address the limitations of classical encryption methods [8], researchers have explored chaotic encryption schemes based on chaotic systems [9]. Chaos was introduced mathematically by Lorenz in 1963 [10], with Robert Matthews pioneering chaotic cryptography using the Logistic map in 1989 [11]. This paved the way for block ciphers based on reverse iterative chaos by T. Habutsu in 1991 [12], sparking research in chaotic encryption for image processing.

Numerous schemes for chaotic image encryption have been proposed, primarily using pixel-level and bit-level permutations. Yang et al. [13] proposed an algorithm using one-dimensional chaotic mapping and compressive sensing, offering hardware scalability. Wen et al. [14] presented a high-quality encryption technique using discrete cosine transform and chaos. Li et al. [15] introduced a color image encryption algorithm employing the Rucklidge system. Despite their effectiveness, these methods face limitations in encryption performance and resistance to attacks.

Hyperchaotic systems, with higher dimensions and nonlinearity, offer enhanced security. Li and Zhang [16] demonstrated a 4D hyperchaotic encryption method, while Demirtas [17] proposed a multi- image encryption approach for grayscale images. However, these methods struggle with multi-channel color image encryption challenges.

To overcome these challenges, we propose RAINBOW, an asymmetric hyperchaotic image encryption technique blending bit-level and pixel-level permutations and employing a bit-plane division strategy. In extensive experiments, RAINBOW outperforms existing methods in security and utility, making it a significant advancement in image encryption technology.

The contributions of this paper can be summarized as follows:

- We introduce RAINBOW, a pioneering hyperchaotic image encryption scheme, distinguished by its focus on multi-channel operations for color images beyond mere grayscale processing. This scheme innovatively merges bit-level and pixel-level permutations and diffusions, powered by a hyperchaotic system to create displacement matrices.
- We adopt bit-plane block segmentation and utilize distance matrices for cross-multi-channel block movements, to achieve a sophisticated blend of bit-level and pixel-level permutations. In the diffusion stage, three-dimensional cross-multi-channel diffusion is implemented, signif- icantly enhancing the scheme's security.
- We conduct comprehensive experiments on a variety of standard test color images of different sizes, which validate the encryption and decryption capabilities of RAINBOW, showcasing its superiority in security performance compared to other contemporary image encryption methods. These tests confirm the advanced security features of our scheme over existing hyperchaotic image encryption approaches.

The remaining sections of this paper are structured as follows: Section 2 provides a detailed explanation of our proposed RAINBOW. Next, Section 3 presents the experimental results and performance analysis of RAINBOW, comparing it to existing hyperchaotic image encryption methods across mul- tiple aspects. Finally, Section 4 concludes the paper.

## 2. The detail of RAINBOW

### 2.1. Encryption Process

In this scheme, we assume that the sender wishes to transmit a color secret image to the receiver through a public channel. The entire process is summarized as follows, comprising two main parts. Initially, the sender encrypts the color plaintext image using a hash value derived from the plaintext image, followed by the generation of a hyperchaotic matrix through a hyperchaotic system, which then produces three displacement matrices for the subsequent permutation. Concurrently, the original image is divided into grayscale images of three color channels, with each color channel further segmented into bit planes. Subsequently, the same bit planes of different colors are merged into bit-level cubes. These cubes and the displacement distance matrices are similarly divided into blocks. Then, the bit- level cubes are permutated block by block according to the displacement distance matrices. Finally, the permutated image is arranged using a pixel-by-pixel diffusion algorithm. The receiver can extract the corresponding plaintext image using the hash value and decryption scheme. The entire process of this scheme is illustrated in Figure 1.

Key and Hyperchaotic Matrix Formation Before generating a three-dimensional hyperchaotic matrix using the LF-NCHM method [18], initial values are generated based on the hash value of the plaintext image. To ensure security, a dynamic hash related to the plaintext image is obtained using the SHA-256 algorithm, indicating that even minimal modifications can lead to significant differences in the outcome. The initial conditions, characterized by high sensitivity, are determined based on this dynamic hash. These conditions are then input into the LF-NCHM to construct the hyperchaotic matrix used in subsequent operations. The specific steps include:

(1) Generating a hash value from the plaintext image.

(2) Dividing the hash into 32 segments to serve as inputs for the LF-NCHM hyperchaotic system.

(3) Iterating through the LF-NCHM nonlinearly coupled hyperchaotic mapping $M \times N$ (original image size) times generates hyperchaotic sequences x and y, as depicted in Equation (1). The

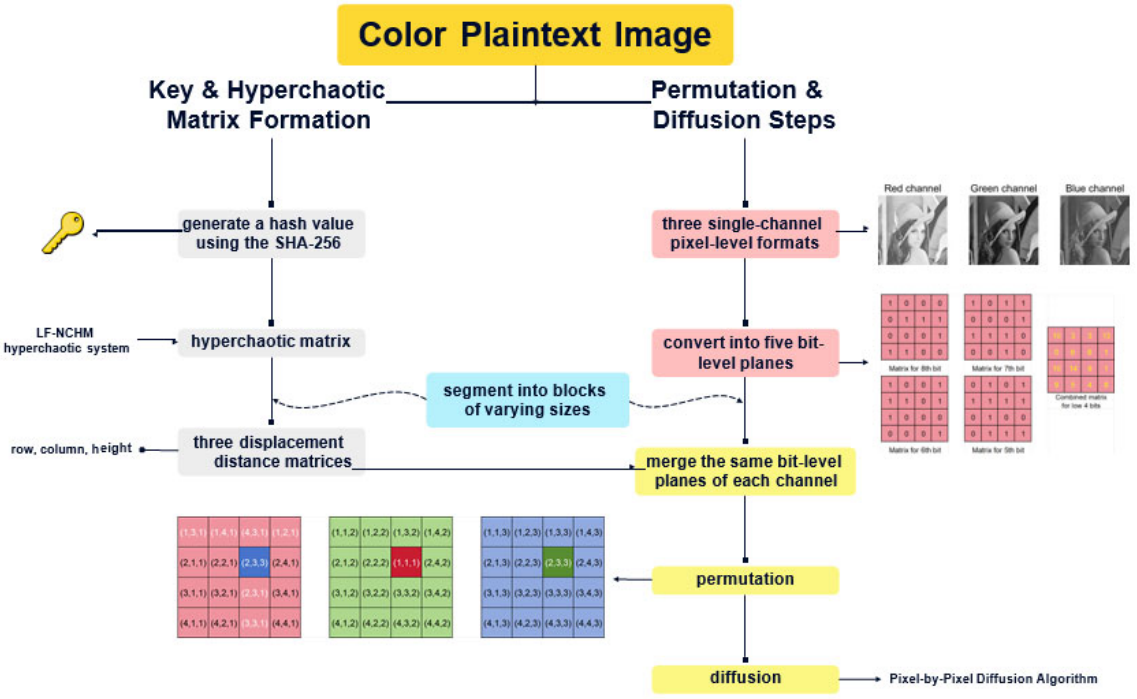


**Fig. 1** The RAINBOW Encryption Scheme

iterative values of x populate a matrix, yielding a hyperchaotic matrix commensurate with the plaintext image's dimensions.

$$\begin{cases} x_n = \mathrm{mod}(\mu * y_{n-1} * (1 - y_{n-1}),1) \\ y_n = \mathrm{mod}(\lambda * sin(\pi * (x_{n-1} + y_{n-1})),1) \end{cases} \quad (1)$$

The parameter μ is restricted to the interval (0,4]. Extensive experimental analysis has indicated the optimal performance of the model at λ = 5. Consequently, λ will be maintained at this value throughout the remainder of the paper.

## 2.2. Permutation Step

During the permutation process, the plaintext image of size $M \times N \times 3$ is con- verted from a color three-channel original image to three single-channel pixel-level formats. The grayscale image of each channel is then converted into five bit-level planes. These planes are di- vided into blocks of varying sizes: bit planes composed of the 8th, 7th, 6th, and 5th bits are divided into blocks of sizes 1 1×1,2×2,4×4,and8×8, respectively. For bit planes combined from the 1st to 4th bits, which contain less information, the block size is set to 16×16 to enhance processing speed. Subsequently, the same bit-level planes of each channel are merged according to the original channel sequence, forming a new $M \times N \times 3$ bit-level cube with identical block sizes. Following this, the five- bit-level cubes undergo permutation based on the block sizes determined earlier, primarily dictated by the distance matrices. Finally, every arranged bit-level cube is decomposed into three color channels, with five bit-level planes of the same color combined to form a single-channel grayscale image. These three color channels are then merged and converted into a pixel-level color image. The plaintext image undergoes both bit-level rearrangement and pixel-level permutation across color channels through the novel arrangement of blocks of varying sizes. The steps are as follows:

(1) After splitting the original plaintext image into three single channels, The grayscale image of each channel is converted into five distinct bit-level planes. As shown in Figure 2, the first plane consists of combinations of the 1st to 4th bits of each pixel in the plaintext image, while the subsequent four planes are formed from the respective 5th to 8th bits.

(2) Based on the information content of each bit, these planes are segmented into blocks of varying sizes. As the 8th bit plane contains more information than any other single bit in each pixel, its block size is 1×1. Accordingly, when the plane's dimensions are multiples of 16, the 7th, 6th, and 5th-bit planes are divided into blocks of sizes 2×2, 4×4, and 8×8, respectively, with zero- padding as necessary. For the aforementioned "combined" plane, the appropriate block size is 16×16 to optimize time efficiency.

(3) To maintain randomness and the relationship with the plaintext image, the previously gener- ated hyperchaotic matrix determines the displacement distance. For operations on the same bit- level cube, the hyperchaotic matrix $(M \times N)$ is divided into blocks of the same size as those in the bit-level plane. The value of each block after division is the sum of the hyperchaotic sequence values within that block in the original matrix. These block values are then sorted: the blocks along each row are traversed, and the sorting indices form the row displacement distance, ensuring the cyclic shift does not exceed boundary values. Similarly, traversing and sorting the block values of each column yield the column displacement distance. The row and column displacement distances form the row and column distance matrices, respectively, sized $(M / BlockSize \times N / BlockSize)$. For the height distance matrix, the sum of block values at the same position in each row and column distance matrix is modulated by 3, ensuring the high displacement distance consistently remains within the 1 to 3 range. This process results in three distance matrices in Figure 3.

(4) Upon establishing the displacement distances, the process unfolds as follows: Initially, the pos- itive direction for displacement is set to rightward, downward, and toward the bottom. Within the partitioned three-dimensional bit-level cube matrix, cyclic shifts are performed sequentially in rows, columns, and height for each block. These blocks are aligned with their correspond- ing positions in the two-dimensional row, column, and height displacement matrices, whose values dictate the shift distances for each dimension. Figure 4 demonstrates this displacement mechanism, utilizing the matrix from Figure 3, with the results shown in Figure 5.

(5) After the permutation, all five bit-level cubes are split back into three color channels and then merged into a single pixel-level color image of size $(M \times N)$, pic, effectively reversing the process described in step 1. This marks the completion of the permutation operation.

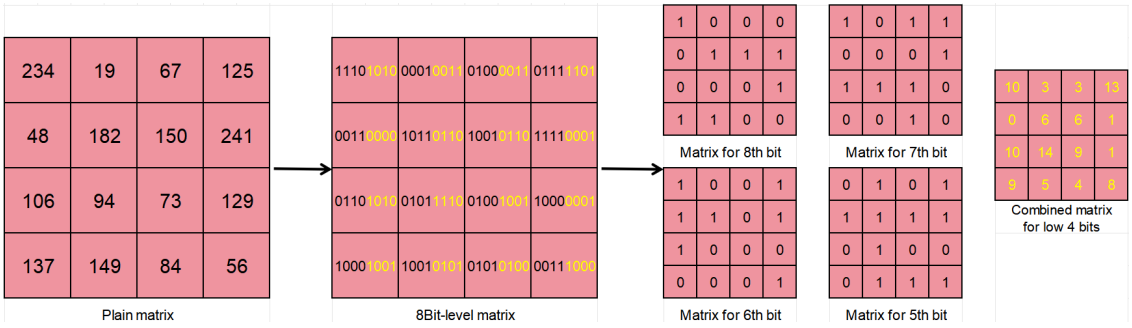


**Fig. 2** The grayscale image of the single-color channel is converted into five bit-level planes.

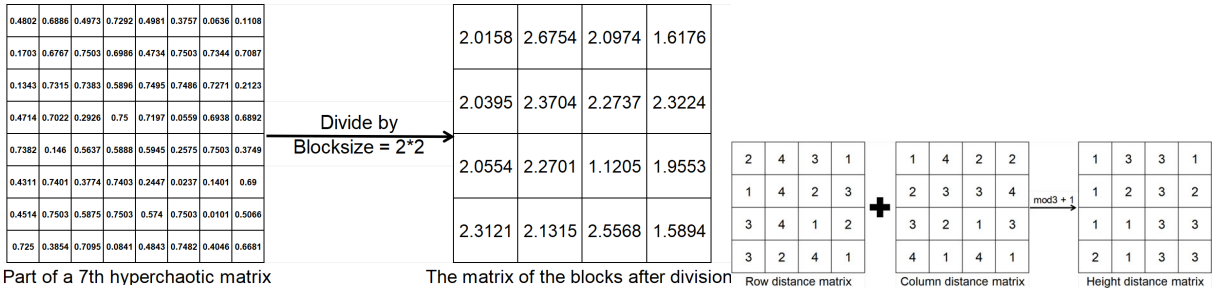


**Fig. 3** An example of a hyperchaotic matrix of the seventh level bit plane divided into blocks, then sorted and modulo to generate the distance matrices in three directions

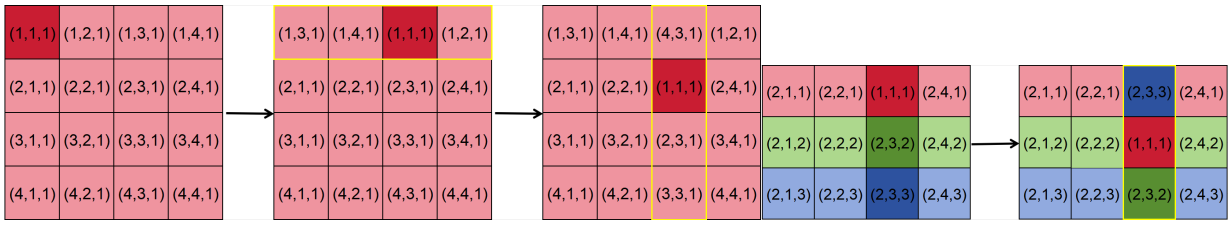


**Fig. 4** i = 1, j = 1, according to the block values of the three displacement distance matrices: row displacement distance = 2, column displacement distance = 1, height displacement distance = 1

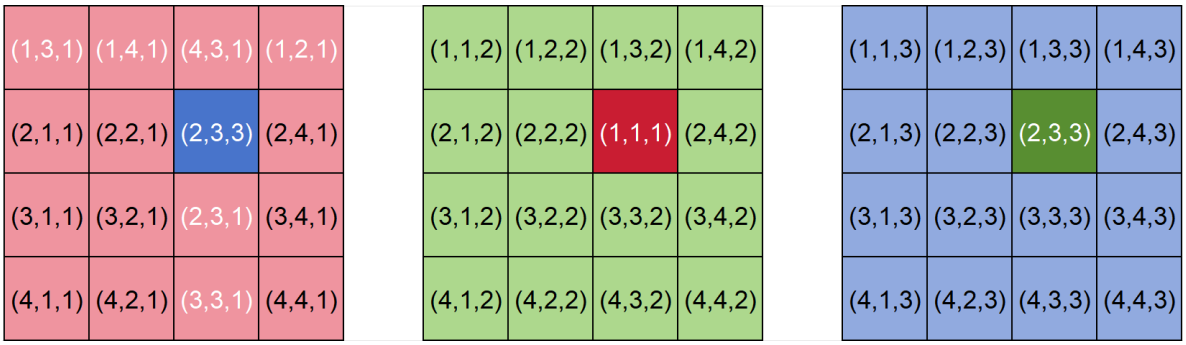


**Fig. 5** Example of permutated red blocks at coordinates (1, 1, 1)

Diffusion Step To minimize the correlation among neighboring pixels within an image, a diffusion process is imperative. There are principally two kinds of diffusion: block diffusion and pixel-pattern-based diffusion. In the algorithm presented in this paper, we have implemented a single-pass pixel-by-pixel diffusion process. This process begins with the pixel located in the bottom-right corner of the three-dimensional image and concludes with the pixel in the bottom-right corner. Throughout this process, diffusion occurs from the bottom to the top layer and progresses from downwards to upwards and right to left. This signifies that the current pixel is influenced by its neighbors to the bottom, to the downwards, and to the right. The operational specifics are as follows:

(1) Upon generating the hyperchaotic sequences, a subsequence is extracted from one of these sequences for diffusion processing. This subsequence, denoted $x(i)$, is subsequently transformed into a two-dimensional sequence $x(i, j)$ with dimensions $M \times N$.

(2) As input for the diffusion process, the permuted image $fig(i, j, c)$ is processed pixel-by-pixel. In this process, each pixel is diffused by incorporating information from its right and lower neighbors, as well as from the corresponding pixel in the subsequent color channel layer. This ensures that the diffusion is spread across the entire image in a bottom-top and down-to-up and right-to-left fashion. The completion of this operation results in the cipher image $pic(i, j, c)$. The specific methodology of the diffusion process is elaborated in Algorithm 1, which follows.

**Algorithm 1** Pixel-by-Pixel Diffusion Process

```
for c = C downto 1 do
    for i = M downto 1 do
      for j = N downto 1 do
          if i = M and j = N and c = C then
              pic(i, j, c) ← fig(i, j, c) ⊕ floor(x(i, j) × 255)
          else if c = C then
              if i = M and j < N then
                  diffu ← fig(i, j, c) ⊕ pic(i, j + 1, c)
              else if i < M and j = N then
                  diffu ← fig(i, j, c) ⊕ pic(i + 1, j, c)
              else if i < M and j < N then
                  diffu ← fig(i, j, c) ⊕ pic(i, j + 1, c)
                  diffu ← diffu ⊕ pic(i + 1, j, c)
              end if
              pic(i, j, c) ← diffu ⊕ floor(x(i, j) × 255)
          else
              if i = M and j = N then
                  diffu ← fig(i, j, c) ⊕ pic(i, j, c + 1)
              else if i = M then
                  diffu ← fig(i, j, c) ⊕ pic(i, j + 1, c)
                  diffu ← diffu ⊕ pic(i, j, c + 1)
              else if j = N then
                  diffu ← fig(i, j, c) ⊕ pic(i + 1, j, c)
                  diffu ← diffu ⊕ pic(i, j, c + 1)
              else
                  diffu ← fig(i, j, c) ⊕ pic(i, j + 1, c)
                  diffu ← diffu ⊕ pic(i + 1, j, c + 1)
                  diffu ← diffu ⊕ pic(i, j, c + 1)
              end if
              pic(i, j, c) ← diffu ⊕ floor(x(i, j) × 255)
          end if
        end for
    end for
end for
```

## 2.3. Decryption Process

In general, the decryption process of the proposed scheme is essentially the inverse of the encryption process. The input for the decryption process is the recipient's private key $u$. The steps are as follows:

1. Decrypt the encrypted hash using the private key $u$.
2. Generate the hyperchaotic sequences $x_0$ and $y_0$ from the decrypted hash, reshape $x_0$ into a two-dimensional array $x(i, j)$ for inverse diffusion, and reshape $x_0$ and $y_0$ into a two-dimensional hyperchaotic matrix for inverse permutation.
3. Execute the inverse diffusion process to obtain an image pic that has only been permuted from the encrypted image fig, utilizing the sequence $x(i, j)$.
4. Implement the inverse permutation process, transforming the hyperchaotic matrix into three two-dimensional matrices representing row, column, and height distances, and then perform the reverse operation of the permutation process on the image pic to retrieve the segmented parts of the original image.
5. Finally, reconstruct these parts into a complete entity and restore it to the decrypted image $P$, which has a size of $M \times N$, the same as the original plaintext image.

## 2.4. Experimental Results

In this section, we detail the experimental setup. The experiments were conducted in the following environment: the software platform was MATLAB R2021a; the hardware configuration included an Inter (R) Core™ i5-11320 CPU @ 3.20 GHz with 16GB of RAM; the operating system was Windows 11 Professional.

## 2.5. Simulation Results

We used two 256 × 256 pixel images ("beans" and "pink") and five 512×512 pixel color images ("Baboon", "house", "Lena", "Peppers", and "sailboat") in our experiments. The simulation results are displayed in Figure 6. The results clearly demonstrate that the encryption process produces images with random characteristics, making the original content unrecognizable to the human eye, thereby effectively proving the efficacy of the encryption technique. Furthermore, the decryption algorithm allows the encrypted images to be efficiently restored to their corresponding original plaintext images. The experimental results confirm that the original information can be precisely recovered without any discrepancies or losses, proving the applicability and effectiveness of the adopted method.


**Fig. 6** The simulation test for the images. The first line presents the plaintext images, and the corresponding ciphertext images are shown below.

# 3. Statistical Results

## 3.1. Correlation Analysis

Typically, in various plaintext images, a strong correlation exists between two adjacent and distinct pixels, necessitating the use of an efficient and secure encryption algorithm to mitigate this correlation. After encrypting the plaintext image, the goal is to achieve a lower correlation coefficient between adjacent pixels in the corresponding ciphertext image. The formula for calculating the correlation between two different adjacent pixels is shown in Equation (2). In our analysis, we selected pixel pairs in three different directions: horizontal, vertical, and diagonal, in both the plaintext and the ciphertext images. Utilizing Equation (2), we calculated the correlation coefficients in these three directions for the plaintext and the corresponding encrypted images. The results of the correlation analysis for the plaintext and encrypted images in these three directions are presented in Table 1. The results of the correlation coefficients are compared across different schemes, as detailed in Table 2. From the data presented, it is evident that the correlation results in the encrypted image are close to zero, indicating a significant reduction in correlation. Furthermore, it is clear that our encryption scheme outperforms the other schemes mentioned earlier.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}}. \quad (2)$$

To demonstrate the randomness in the simulation tests, nearly 10,000 pairs of pixels were randomly selected from three directions of the image. Analysis of their distribution plots clearly indicates that, after encryption, the values of adjacent pixels in all three directions are completely random. This

demonstrates a reliable defense against statistical attacks, implying that any attempt by attackers to extract statistical information from the encrypted images would be futile. In summary, our scheme exhibits strong performance through statistical analysis.

**Table 1.** Correlation Coefficients of three direc- tions for Various Standard Test Images.

| | **Horizontal** | **Vertical** | **Diagonal** |
|---|---|---|---|
| baboon | -0.0023 | -0.0011 | 0.0011 |
| beans | -0.0023 | 0.0026 | 0.0008 |
| house | 0.0039 | **0.00007** | -0.0007 |
| Lena | **-0.0002** | -0.0024 | 0.0026 |
| peppers | -0.0014 | 0.0005 | 0.0009 |
| pink | 0.0015 | 0.0012 | -0.0018 |
| sailboat | 0.0023 | -0.0015 | **-0.0001** |

**Table 2.** Correlation Coefficients Comparison of Encrypted Lena Image Across Different Schemes.

| | **Horizontal** | **Vertical** | **Diagonal** |
|---|---|---|---|
| Lena | 0.9605 | 0.9764 | 0.9478 |
| **Proposed** | **-0.0002** | **-0.0024** | **0.0026** |
| [19] | 0.0014 | -0.0015 | 0.0079 |
| [20] | 0.0105 | -0.0023 | 0.0052 |
| [21] | -0.0014 | 0.0038 | -0.0083 |
| [22] | -0.0042 | -0.0028 | 0.0027 |
| [23] | 0.0076 | 0.0024 | 0.0043 |

### 3.2. Sensitivity Analysis

Key sensitivity is a critical attribute of image encryption schemes, reflecting their robustness against brute-force attacks. The nature of key generation dictates that even minimal alterations in the plain-text image result in a completely different encrypted output. Additionally, slight variations in hash values can generate distinct ciphertext images, making it impossible to retrieve the plaintext from the ciphertext when the key undergoes minor modifications. This high sensitivity to input parameters is a notable characteristic of hyperchaotic systems, where even the slightest changes can lead to the afore- mentioned outcomes. Figure 7 depicts the original image of Lena, its corresponding encrypted image, the image decrypted with the correct key, and three images decrypted with keys that have a single-bit change. The incorrect decryption results demonstrate that even a minor alteration in the hash value prevents the correct decryption of the image. Hence, this new scheme exhibits strong key sensitivity.

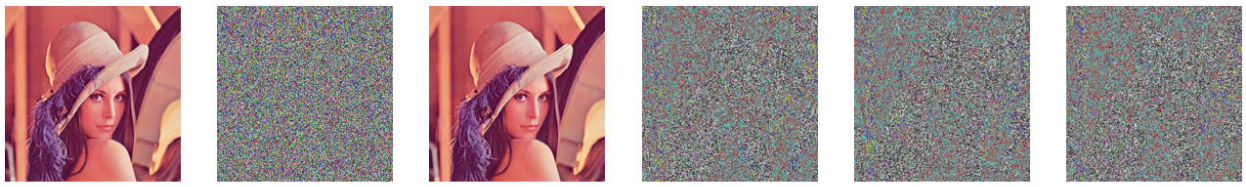


**Fig. 7** Testing results for key sensitivity.

## 4. Information Analysis

### 4.1. Information Entropy

Information entropy is a crucial metric that demonstrates the randomness of an image and measures the effectiveness of an encryption scheme, revealing the level of obfuscation in the image. The formula for calculating information entropy is as follows:

$$H = -\sum_{m=1}^{L} p\,(m)\,log\quad_2 p(m). \tag{3}$$

In this formula, $p(m)$ represents the probability of symbol $m$, where $m$ originates from the information source. $H(m)$ denotes the value of entropy, and $L$ represents the number of gray levels in the image. The theoretical value of information entropy $H$ is 8. An increase in image entropy indicates higher uncertainty, making it more difficult for attackers to extract information from the image. Table 3 displays the entropy values for seven encrypted images. Table 4 showcases our encryption method's excellence, achieving near-ideal entropy close to 8 across color channels and in mean value—indicative of superior data obfuscation compared to existing techniques. From these values, it is evident that after encryption, the entropy of all seven cipher images is very close to the ideal theoretical value and significantly different from their corresponding plaintext images, signifying that the algorithm proposed here effectively randomizes the pixels of digital images.

**Table 3.** Entropy Values of RGB Channels for Various Standard Test Images.

| | **Red** | **Green** | **Blue** |
|---|---|---|---|
| baboon | 7.9973 | 7.9971 | 7.9973 |
| beans | 7.9969 | 7.9968 | 7.9974 |
| house | 7.9973 | 7.9972 | 7.9968 |
| Lena | 7.9972 | 7.9967 | 7.9973 |
| peppers | 7.9971 | 7.9972 | 7.9972 |
| pink | 7.9972 | 7.9971 | 7.9973 |
| sailboat | 7.9973 | 7.9974 | 7.9966 |

**Table 4.** Entropy Comparison of Encrypted Lena Image Across Different Schemes.

| | **Red** | **Green** | **Blue** | **Mean** |
|---|---|---|---|---|
| Lena | 7.2531 | 7.5940 | 6.9684 | 7.2718 |
| **Proposed** | **7.9972** | **7.9967** | **7.9973** | **7.9971** |
| [19] | 7.9972 | 7.9965 | 7.9963 | 7.9967 |
| [24] | N/A | N/A | N/A | 7.9924 |
| [25] | 7.9972 | 7.9973 | 7.9966 | 7.9970 |
| [26] | N/A | N/A | N/A | 7.9856 |
| [27] | N/A | N/A | N/A | 7.9916 |

### 4.2. Robustness Analysis

In assessing the robustness of the encryption algorithm for three-dimensional color images, the chi-square (χ2) test was employed as a statistical measure to evaluate the distribution of pixel values. The chi-square test quantifies the discrepancies between the observed frequencies of pixel values and the frequencies expected in a uniformly distributed image. For a three-dimensional image with dimensions M × N × O, the test is performed by first flattening the image into a one-dimensional array and then calculating the χ2 value as:

$$\chi^2 = \sum_{i=0}^{255} \frac{(q_i - \bar{q})^2}{\bar{q}} \tag{4}$$

where $q_i$ is the observed frequency of the $i$-th pixel value, and $\overline{q}$ is the expected frequency of each pixel value for a perfectly uniform distribution, computed as $(M \times N \times O)/256$ given the 256 possible intensity values for each color channel. The $\chi^2$ value for each channel is calculated and then averaged to produce a composite metric representing the overall encryption quality.The results presented in Table 5 illustrate that the average $\chi^2$ value for each cryptographic histogram test result is less than the critical value of $\chi^2_{0.05}(255) = 293.24793$ at a significance level of $\alpha = 0.05$. Such findings suggest that the cryptographic images exhibit a uniform distribution across all channels. This indicates that the proposed algorithm successfully obscures any predictable patterns in the histogram, which could otherwise be leveraged in a statistical attack. The uniform distribution across the RGB

channels thus reinforces the algorithm's capability to resist histogram-based statistical attacks effectively.

### 4.3. MSE and PSNR

In the assessment of three-dimensional color images, the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) serve as key metrics for evaluating the accuracy of images before and after encryption. These metrics are particularly adapted for three-dimensional images by considering the color channels and depth information. The MSE is calculated as:

$$\mathrm{MSE} = \frac{1}{M \times N \times O} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{O} (P(i,j,k) - C(i,j,k))^2.$$

And the PSNR is given by:

$$\mathrm{PSNR} = 20\, log_{10} \left( \frac{I_{\max}}{\sqrt{\mathrm{MSE}}} \right),$$

where $M$, $N$, and $O$ represent the dimensions of the image; $P$ is the original plaintext image; $C$ is the cipher image; and Imax is the maximum pixel value which, for color images, is typically 255.

Upon applying these metrics to three-dimensional color images, the resulting data suggests that the proposed encryption algorithm induces significant distortion as reflected by high MSE values, and the encrypted images exhibit substantial deviation from the original ones. This degree of alteration enhances the security of the encrypted images, as evidenced by low PSNR values, which indicate that the encrypted images have a high level of noise-like characteristics, making them resistant to decryption attempts without the proper key. The tabulated results in Table 5 corroborate the efficacy of the encryption process, showcasing the algorithm's ability to secure image data effectively against unauthorized access and statistical analysis.

**Table 5.** Test Results of Multiple Metrics for Various Standard Test Images.

| | NPCR | UACI | $\chi^2$ | MSE | PSNR |
|---|---|---|---|---|---|
| baboon | 99.6092 | 33.4675 | 250.1536 | 8262.1257 | 8.9598 |
| beans | 99.6093 | 33.4631 | 266.2473 | 8852.2793 | 8.6602 |
| house | 99.6086 | 33.4632 | 257.4869 | 9126.3433 | 8.5278 |
| Lena | 99.6097 | 33.4628 | 261.8125 | 8857.6551 | 8.6576 |
| peppers | 99.6101 | 33.4611 | 256.4296 | 10060.4848 | 8.1046 |
| pink | 99.6065 | 33.4680 | 249.6562 | 8500.2391 | 8.8364 |
| sailboat | 99.6094 | 33.4618 | 258.2500 | 10048.7960 | 8.1096 |

**Table 6.** NPCR and UACI Comparison of Encrypted Lena Image.

| | NPCR | Gap | UACI | Gap |
|---|---|---|---|---|
| **proposed** | **99.6097** | **0.0003** | **33.4628** | **0.0007** |
| [21] | 99.6063 | 0.0031 | 33.4607 | 0.0028 |
| [22] | 99.6110 | 0.0016 | 33.4606 | 0.0029 |
| [25] | 99.6287 | 0.0193 | 30.3432 | 3.1203 |
| [28] | 99.5968 | 0.0126 | 33.4747 | 0.0112 |
| [29] | 99.6112 | 0.0018 | 33.4910 | 0.0275 |
| [30] | 99.6257 | 0.0163 | 33.4768 | 0.0133 |

### 4.4. NPCR and UACI

To thwart potential attackers who might attempt to pinpoint vulnerabilities in an encryption scheme by slightly altering a single pixel of the plaintext image and analyzing the differences between the original and the newly encrypted images, image encryption algorithms must respond sensitively to even the slightest change, such as a single-bit modification. Various methods have been proposed to

test this property, employing metrics like the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI), with their theoretical values being 99.6094% and 33.4635%, respectively. To compute these metrics, the plaintext image is initially encrypted, resulting in an image named $c$, and then a single pixel in the plaintext image is altered and re-encrypted to obtain image $C_0$. The experimental results for these metrics are derived by inputting these two images into the formulas presented in Equations (5) to (7). This procedure is repeated 20 times with different randomly selected pixels altered each time. The average results of these two metrics for seven images after 20 iterations are shown in Table 5. Compared to the theoretical values, the experimental results demonstrate robust resistance to differential attacks

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j)}{M \times N} \times 100\%, \tag{5}$$

$$D(i,j) = \begin{cases} 1 & \text{if } C(i,j) \neq C_0(i,j) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

$$\mathrm{UACI} = \frac{1}{M \times N}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|C(i,j) - C_0(i,j)|}{255} \times 100\%. \tag{7}$$

In these formulas, $M \times N$ represents the size of the image, and the symbol × denotes the signum function, while mi signifies the average absolute difference between two elements. Additionally, Table 6 displays the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) values for this study alongside results from other references. A comparison of the results clearly indicates that, for the NPCR performance of our method and the UACI results, our algorithm's values are closer to the theoretical values than any other algorithm. Considering these aspects, the algorithm proposed in this paper demonstrates robust performance in resisting differential attacks.

## 5. Conclusion

Most favored image encryption schemes either consider only a subset of bit levels or the full 8-bit data, with scant attention to the interrelation between different bit levels and their division into separate planes. Furthermore, for color images, most methods do not address the connections between different color layer planes, which limits the diversity of encrypted image data and ultimately negatively impacts encryption effectiveness.

To address these issues, we present RAINBOW, a novel color image encryption and decryption scheme based on asymmetric encryption and hyperchaotic systems. RAINBOW's key feature is its ability to group different bit-level data for permutation: the top 4 bits, containing a larger quantity of information, each form a separate plane, while the lower 4 bits are combined into a single plane, yielding a total of 5 planes. Subsequently, permutation is conducted on planes of the same bit level within a three-dimensional space. During diffusion, a hyperchaotic sequence is employed for enhanced efficiency. Specifically, for pixels in three-dimensional space, the process begins from the bottom-right corner, moving from bottom to top, downwards to upwards, and right to left. After this process, pixels become associated with those in the three directions of the bottom, down, and right sides, achieving a more complex and secure correlation.

Extensive experimental results indicate that the proposed RAINBOW holds significant advan- tages in resisting common attacks such as statistical analysis and chosen-plaintext attacks, offering a higher degree of security. One limitation of RAINBOW is its time-consuming nature; future work will investigate how to accelerate the process and further consider its application to hardware.

## References


[1] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," Entropy, vol. 23, no. 3, 2021. [Online]. Available: https://www.mdpi. com/1099-4300/23/3/341

[2] C. L. Chowdhary, P. V. Patel, K. J. Kathrotia, M. Attique, K. Perumal, and M. F. Ijaz, "Analytical study of hybrid techniques for image encryption and decryption," Sensors, vol. 20, no. 18, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5162

[3] P. T. Akkasaligar and S. Biradar, "Selective medical image encryption using dna cryptography," Information Security Journal: A Global Perspective, vol. 29, no. 2, pp. 91–101, 2020.

[4] R. Nithya and D. Dhanasekaran, "Novel dominant color subband image encryption in visual sen- sor network for smart military surveillance system," Traitement du Signal, vol. 39, pp. 951–960, 06 2022.

[5] G. Bortsova, C. González-Gonzalo, S. C. Wetstein, F. Dubost, I. Katramados, L. Hogeweg, B. Liefers, B. van Ginneken, J. P. Pluim, M. Veta, C. I. Sánchez, and M. de Bruijne, "Adversarial attack vulnerability of medical image analysis systems: Unexplored factors," Medical Image Analysis, vol. 73, p. 102141, 2021. [Online]. Available: https://www.sciencedirect.com/science/ article/pii/S1361841521001870

[6] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," Information Sciences, vol. 546, pp. 1063–1083, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0020025520309427

[7] H. Wen and Y. Lin, "Cryptanalysis of an image encryption algorithm using quantum chaotic map and dna coding," Expert Systems with Applications, vol. 237, p. 121514, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742302016X

[8] M. Habek, Y. Genc, N. Aytas, A. Akkoc, E. Afacan, and E. Yazgan, "Digital image encryp- tion using elliptic curve cryptography: A review," in 2022 International Congress on Human- Computer Interaction, Optimization and Robotic Applications (HORA), 2022, pp. 1–8.

[9] X. Wang, S. Chen, and Y. Zhang, "A chaotic image encryption algorithm based on random dynamic mixing," Optics & Laser Technology, vol. 138, p. 106837, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030399220314705.

[10] T. Li, B. Du, and X. Liang, "Image encryption algorithm based on logistic and two-dimensional lorenz," IEEE Access, vol. 8, pp. 13 792–13 805, 2020.

[11] R. Matthews, "On the derivation of a "chaotic" encryption algorithm," Cryptologia, vol. 13, no. 1, pp. 29–42, 1989.

[12] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in Advances in Cryptology — EUROCRYPT '91, D. W. Davies, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 127–140.

[13] C. Yang, P. Pan, and Q. Ding, "Image encryption scheme based on mixed chaotic bernoulli measurement matrix block compressive sensing," Entropy, vol. 24, no. 2, 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/2/273

[14] H. Wen, L. Ma, L. Liu, Y. Huang, Z. Chen, R. Li, Z. Liu, W. Lin, J. Wu, Y. Li, and C. Zhang, "High-quality restoration image encryption using dct frequency-domain compression coding and chaos," Scientific Reports, vol. 12, 10 2022.

[15] Z. Li, C. Peng, W. Tan, and L. Li, "A novel chaos-based color image encryption scheme using bit-level permutation," Symmetry, vol. 12, no. 9, 2020. [Online]. Available: https: //www.mdpi.com/2073-8994/12/9/1497

[16] T. Li and D. Zhang, "Hyperchaotic image encryption based on multiple bit permutation and diffusion," Entropy, vol. 23, no. 5, 2021. [Online]. Available: https://www.mdpi.com/ 1099-4300/23/5/510.

[17] M. Demirtaş, "A novel multiple grayscale image encryption method based on 3d bit- scrambling and diffusion," Optik, vol. 266, p. 169624, 2022. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S0030402622009159.

[18] W. Hou, S. Li, J. He, and Y. Ma, "A novel image-encryption scheme based on a non-linear cross-coupled hyperchaotic system with the dynamic correlation of plaintext pixels," Entropy, vol. 22, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/7/779

[19] W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, and A. Aboshousha, "Color image encryption through chaos and kaa map," IEEE Access, vol. 11, pp. 11 541–11 554, 2023.

[20] K. Kumar, S. Roy, U. Rawat, and S. Malhotra, "Iehc: An efficient image encryption technique using hybrid chaotic map," Chaos, Solitons & Fractals, vol. 158, p. 111994, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077922002041.

[21] Q. Lai and Z. Chen, "Grid-scroll memristive chaotic system with application to image encryption," Chaos, Solitons & Fractals, vol. 170, p. 113341, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077923002424.

[22] W. Hao, T. Zhang, X. Chen, and X. Zhou, “A hybrid neqr image encryption cryptosystem using two-dimensional quantum walks and quantum coding,” Signal Processing, vol. 205, p. 108890, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0165168422004297.

[23] X. Kong, F. Yu, W. Yao, C. Xu, J. Zhang, S. Cai, and C. Wang, “A class of 2n+1 dimensional simplest hamiltonian conservative chaotic systems and fast image encryption schemes,” Applied Mathematical Modelling, vol. 125, pp. 351–374, 2024. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S0307904X23004456

[24] B. Jasra and A. Hassan Moon, “Color image encryption and authentication using dynamic dna encoding and hyper chaotic system,” Expert Systems with Applications, vol. 206, p. 117861, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422011162.

[25] W. Alexan, M. ElBeltagy, and A. Aboshousha, “Rgb image encryption through cellular automata, s-box and the lorenz system,” Symmetry, vol. 14, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2073-8994/14/3/443

[26] M. Gabr, H. Younis, M. Ibrahim, S. Alajmy, I. Khalid, E. Azab, R. Elias, and W. Alexan, “Application of dna coding, the lorenz differential equations and a variation of the logistic map in a multi-stage cryptosystem,” Symmetry, vol. 14, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2073-8994/14/12/2559

[27] M. Hui, X. Liu, S. Zhu, and J. Cao, “Event-triggered impulsive cluster synchronization of coupled reaction–diffusion neural networks and its application to image encryption,” Neural Networks, vol. 170, pp. 46–54, 2024. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S0893608023006445

[28] D. Wei, M. Jiang, and Y. Deng, “A secure image encryption algorithm based on hyper-chaotic and bit-level permutation,” Expert Systems with Applications, vol. 213, p. 119074, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422020929

[29] Y. Wu, L. Zhang, S. Berretti, and S. Wan, “Medical image encryption by content-aware dna com- puting for secure healthcare,” IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 2089–2098, 2023.

[30] Y. Peng, Z. Lan, K. Sun, and W. Xu, “A simple color image encryption algorithm based on a discrete memristive hyperchaotic map and time-controllable operation,” Optics & Laser Technology, vol. 165, p. 109543, 2023. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S003039922300436X