entropy

MDPI

Article

Color Image Encryption Based on 3D-SBFCM with Dynamic Rectangular Partitioning and Dynamic S-Box Substitution

Ting Wang [1,*], Xiaoyan Yang [1], Bin Ge [2], Chenxing Xia [2,3] and Houyue Wu [4]

[1] School of Information Engineering, Huainan Union University, Huainan 232038, China; yangxiaoyan@hnuu.edu.cn
[2] School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China; bge@aust.edu.cn (B.G.); cxxia@aust.edu.cn (C.X.)
[3] State Key Laboratory of Digital Intelligent Technology for Unmanned Coal Mining, Anhui University of Science and Technology, Huainan 232001, China
[4] School of Computer and Information Engineering, Bengbu University, No. 1866 Caoshan Road, Bengbu 233030, China; why@bbc.edu.cn
* Correspondence: wangting@hnuu.edu.cn

Abstract

Existing chaos-based color image encryption algorithms still face several challenges, including insufficient dynamical complexity of low-dimensional chaotic maps, residual boundary regularity caused by fixed block partitioning, and limited diffusion among RGB channels. To address these issues, this paper proposes a color image encryption algorithm based on a three-dimensional sine-bilinear fully coupled chaotic map (3D-SBFCM). The proposed map integrates sinusoidal modulation, linear coupling, and bilinear cross-coupling within a mod-1 mapping framework, thereby improving the complexity and pseudorandomness of the generated chaotic sequences. In addition, a residual-feasibility-constrained dynamic rectangular partitioning mechanism is developed to generate reversible non-uniform image blocks and reduce the structural regularity associated with fixed-size partitioning. Based on this partitioning structure, inter-block permutation among same-size blocks and intra-block two-dimensional permutation are performed to weaken both global and local spatial correlations. Plaintext-related initialization, dynamic S-box substitution, and forward-backward cross-channel diffusion are further incorporated into the overall permutation-diffusion framework to enhance plaintext sensitivity, nonlinear confusion, and perturbation propagation across RGB channels. Experimental results demonstrate that the proposed algorithm effectively conceals the statistical characteristics of plaintext images, with information entropy values higher than 7.999 for all color channels and NPCR/UACI values close to their theoretical expectations. The algorithm also shows satisfactory robustness against cropping and noise attacks. These results indicate that the proposed method provides an effective and secure solution for color image encryption.

Keywords: color image encryption; chaotic map; hierarchical scrambling; dynamic S-box; cross-channel diffusion

Check for updates

Academic Editor: Andrea Murari

Received: 6 May 2026
Revised: 3 June 2026
Accepted: 7 June 2026
Published: 9 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Images are characterized by high redundancy, strong spatial correlation, and large data volume [1,2], which makes image encryption fundamentally different from conventional text encryption. A secure image cipher should conceal visual information, suppress statistical features, and reduce correlations among neighboring pixels. Owing to their

sensitivity to initial conditions, ergodicity, aperiodicity, and pseudo-random behavior, chaotic systems have been widely used in image encryption [3–6]. Early studies mainly adopted low-dimensional maps, such as the logistic map [7,8], sine map [9], and Henon map [10,11]. Although these maps are easy to implement, their simple structures usually lead to narrow chaotic intervals, limited key spaces, uneven sequence distributions, and weak resistance to statistical analysis, especially in color image encryption requiring strong plaintext sensitivity and inter-channel diffusion.

To improve classical low-dimensional maps, researchers have introduced nonlinear reconstruction, parameter extension, hybrid coupling, and bit-level perturbation. Sun et al. [12] combined Logistic and Sine maps with cellular automata for color image encryption. Dua et al. [13] introduced fractional-order operations into a cosine-based map to enlarge the key space. Li et al. [14] proposed an enhanced logistic modular map using modular and exponential nonlinearities. Zhao et al. [15] constructed a Henon-Sine coupled map, whereas Sharma et al. [16] employed XOR-based bit rearrangement to extend the chaotic behavior of the Henon map. These methods improve sequence complexity to some extent; however, many of them still suffer from limited state interaction, making it difficult to provide sufficient global disturbance for complex color images.

High-dimensional and hyperchaotic systems have also attracted considerable attention. Zhang et al. [17] developed a four-dimensional hyperchaotic system based on the Chen system and combined it with SHA-384 to increase plaintext dependence. Alexan et al. [18] used a five-dimensional chaotic system and nonlinear transformations to expand the key domain. Sun et al. [19] proposed a six-dimensional non-degenerate discrete hyperchaotic system with compressed sensing and plaintext-related scrambling. Kaur et al. [20] applied a seven-dimensional chaotic system to lightweight biomedical image protection. Compared with low-dimensional maps, these systems generally exhibit richer dynamics and stronger parameter sensitivity. Nevertheless, high-dimensional continuous systems often involve multiple parameters and numerical integration, resulting in higher computational cost and implementation complexity. This limits their deployment in real-time or resource-constrained scenarios, such as mobile devices and IoT edge terminals [21].

Meanwhile, chaos-based encryption has been combined with DNA coding, blockchain, artificial intelligence, and other techniques. Chen et al. [22] integrated an improved Arnold map, dynamic DNA encoding, and a dual hyperchaotic system to reduce pixel correlation. Kumaran et al. [23] combined DNA operations with elliptic curve cryptography for medical image encryption. Neela et al. [24] proposed a blockchain-driven private-key generation scheme, while Inam et al. [25] introduced blockchain-assisted signature verification into chaotic medical image transmission. Ahn and Hong [26] designed a quantum GAN-based key generator, and Nair et al. [27] combined hyperchaotic maps, memristive maps, and convolutional neural networks. These methods can enhance security to some extent, but they may also increase computational complexity, storage overhead, communication cost, or implementation difficulty. The review studies by Li et al. [28] and Gao et al. [29] also pointed out that chaos-based encryption algorithms should not only focus on statistical randomness, but also comprehensively consider the balance among security, complexity, implementation efficiency, and practical application scenarios.

In addition to chaotic sequences themselves, permutation and diffusion structures are also important factors affecting the security of image encryption. Many image encryption algorithms adopt a permutation-diffusion structure, where permutation is used to change pixel positions and diffusion is used to change pixel values. Fixed-size block scrambling has the advantages of simple implementation and high computational efficiency. However, its block boundaries are usually located at fixed row and column intervals. For example, 8×8 or 16×16 block partitioning forms a regular grid structure. Even after inter-block per-

mutation, such fixed grid boundaries may still retain certain structural regularity, which is referred to in this paper as residual boundary regularity caused by fixed block partitioning. Therefore, relying only on fixed-size block partitioning may be insufficient to fully destroy the spatial structure of an image.

Moreover, some color image encryption algorithms process the R, G, and B channels separately, or introduce only weak inter-channel diffusion. When a slight pixel change occurs in one channel, the perturbation may not sufficiently propagate to the other channels or the entire cipher image, thereby affecting the algorithm's resistance to differential attacks. Therefore, color image encryption algorithms need to jointly consider non-uniform spatial scrambling, nonlinear substitution, and cross-channel diffusion.

In summary, existing methods still have the following limitations. First, some low-dimensional chaotic maps have simple structures and high efficiency, but their dynamical complexity and state coupling capability are insufficient. Second, high-dimensional or continuous hyperchaotic systems exhibit richer dynamical behaviors, but they usually involve higher computational complexity and implementation cost. Third, hybrid schemes based on DNA coding, blockchain, artificial intelligence, and other mechanisms can enhance security, but they may introduce additional overhead. Fourth, fixed-size block partitioning tends to produce regular boundaries and may retain grid-like structural regularity. Fifth, the diffusion capability among RGB channels still needs to be further strengthened.

Based on these considerations, this paper adopts a three-dimensional sine-bilinear fully coupled chaotic map to generate pseudorandom sequences and constructs the encryption algorithm from three aspects: permutation, substitution, and diffusion. Specifically, dynamic rectangular block partitioning is used to reduce the residual boundary regularity caused by fixed block partitioning; a dynamic S-box is introduced to enhance nonlinear confusion; and a forward-backward cross-channel diffusion mechanism is designed to strengthen disturbance propagation among RGB channels. The main contributions are summarized as follows:

(1) A new 3D-SBFCM is constructed by integrating sinusoidal modulation, linear coupling, and bilinear cross-coupling into a discrete mod-1 map, thereby improving the complexity and pseudorandomness of the generated chaotic sequences.

(2) A residual-feasibility-constrained dynamic rectangular partitioning strategy is proposed to generate reversible non-uniform image blocks and reduce the boundary regularity caused by fixed-size partitioning.

(3) Based on the proposed partitioning structure, same-size inter-block permutation and intra-block two-dimensional permutation are performed to weaken global and local spatial correlations while maintaining block-boundary compatibility.

(4) Plaintext-related initialization, dynamic S-box substitution, and forward–backward cross-channel diffusion are incorporated into the permutation-diffusion framework to enhance plaintext sensitivity, nonlinear confusion, and RGB-channel perturbation propagation.

2. Construction and Analysis of the Proposed 3D-SBFCM

This section presents the construction of the proposed three-dimensional sine-bilinear fully coupled chaotic map (3D-SBFCM) and then analyzes its dynamical behavior and pseudorandomness.

2.1. Classical One-Dimensional Mod-1 Map

The classical one-dimensional mod-1 map is a nonlinear discrete system defined by the modulo operation. It is characterized by structural simplicity, a bounded state space, and ease of implementation. The mathematical form of the map is given in Equation (1).

$$x_{n+1} = mod(mx_n, 1) \tag{1}$$

where x_n denotes the system state and m is the control parameter. When $m > 1$, the system generates complex trajectories through stretching and folding. However, since this system contains only a single state variable and has a low dynamical dimension, there are still certain limitations regarding state ergodicity and complexity.

2.2. Construction of the Proposed Three-Dimensional Sine-Bilinear Fully Coupled Chaotic Map

To enhance the complexity and ergodicity of discrete chaotic systems, two additional state variables are introduced within the framework of the classical mod-1 map, thereby extending the original one-dimensional system to a three-dimensional one. In addition, a sinusoidal term is incorporated into the state equations to introduce periodic nonlinear perturbations, while linear coupling and bilinear cross-coupling terms are employed to strengthen the interactions among the state variables. Based on this design, a three-dimensional sine-bilinear fully coupled chaotic map (3D-SBFCM) is proposed, and its mathematical form is given in Equation (2).

$$\begin{cases} x_{n+1} = \bmod(mx_n + \alpha \sin(2\pi y_n) + \beta z_n + \gamma y_n z_n, 1) \\ y_{n+1} = \bmod(my_n + \alpha \sin(2\pi z_n) + \beta x_{n+1} + \gamma z_n x_{n+1}, 1) \\ z_{n+1} = \bmod(mz_n + \alpha \sin(2\pi x_{n+1}) + \beta y_{n+1} + \gamma x_{n+1} y_{n+1}, 1) \end{cases} \tag{2}$$

In this map, m controls the linear expansion strength and serves as a key factor in chaos generation; α determines the strength of sinusoidal modulation and enhances the irregularity and sensitivity of state evolution; β denotes the linear coupling strength; and γ denotes the bilinear cross-coupling strength. Owing to the synergistic effects of these nonlinear mechanisms, the system exhibits complex chaotic dynamics, thus laying the foundation for subsequent keystream generation.

2.3. Chaotic Analysis of the Proposed 3D-SBFCM

To verify the chaotic properties of the proposed 3D-SBFCM, phase portraits, Lyapunov exponents, permutation entropy, the 0-1 test, and the NIST SP 800-22 test are employed to perform a comprehensive evaluation in terms of trajectory distribution, sensitivity to initial conditions, time-series complexity, and statistical randomness.

2.3.1. Phase Portrait

Phase portraits visually illustrate the distribution of system trajectories in state space and therefore serve as useful tools for evaluating ergodicity and attractor structure. If a system exhibits pronounced chaotic behavior, its trajectories are typically distributed more widely throughout the phase space, and its attractor structure tends to be more complex [30].

Figure 1 shows the two-dimensional phase portraits of the 3D-SBFCM and compares them with the three-dimensional phase portraits of the Chen and Lorenz systems. In the simulations, the parameters of the 3D-SBFCM were set as $m = 3.7$, $\alpha = 0.5$, $\beta = 0.3$, and $\gamma = 0.2$, with initial conditions $x_0 = 0.1234$, $y_0 = 0.2345$, and $z_0 = 0.3456$.

As shown in Figure 1a–c, the state points of the 3D-SBFCM are relatively uniformly distributed over different projection planes, without obvious periodic clustering, indicating good ergodicity. As shown in Figure 1d–f, compared with the localized attractor trajectories of the Chen and Lorenz systems, the proposed 3D-SBFCM displays a more uniform distribution within the bounded unit state space.

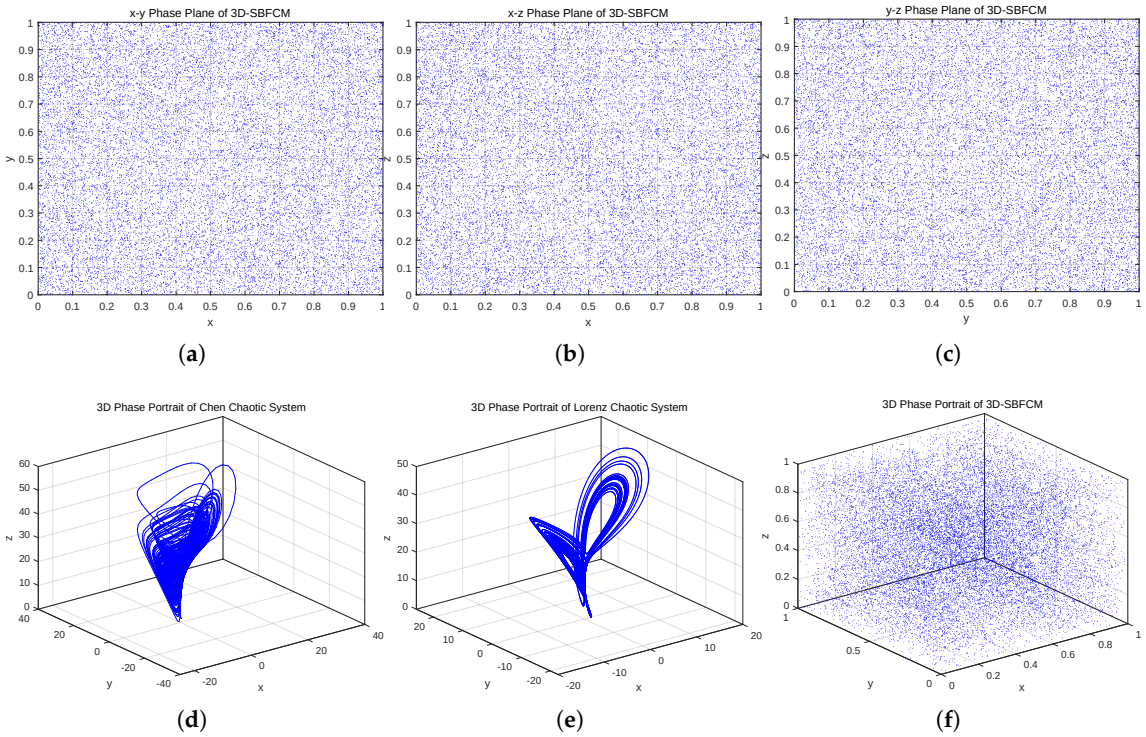


Figure 1. Phase portraits of chaotic systems. (**a**) x–y phase plane of the 3D-SBFCM system; (**b**) x–z phase plane of the 3D-SBFCM system; (**c**) y–z phase plane of the 3D-SBFCM system; (**d**) three-dimensional phase portrait of the Chen system; (**e**) three-dimensional phase portrait of the Lorenz system; and (**f**) three-dimensional phase portrait of the 3D-SBFCM system.

2.3.2. Lyapunov Exponents

The Lyapunov exponent (LE) provides a quantitative measure of a nonlinear system's sensitivity to initial conditions. In general, chaotic systems have one positive LE, whereas hyperchaotic systems have at least two positive LEs [31].

Figure 2 shows the Lyapunov exponents of the 3D-SBFCM, the Chen system, and the Lorenz system. The results indicate that the proposed system exhibits two positive Lyapunov exponents in most of the tested parameter ranges. This confirms that the 3D-SBFCM exhibits stable hyperchaotic behavior and good parameter robustness. By contrast, the parameter intervals over which the Chen and Lorenz systems exhibit positive LEs are narrower and show stronger fluctuations, indicating that their chaotic dynamics are less stable under parameter variations than those of the 3D-SBFCM.

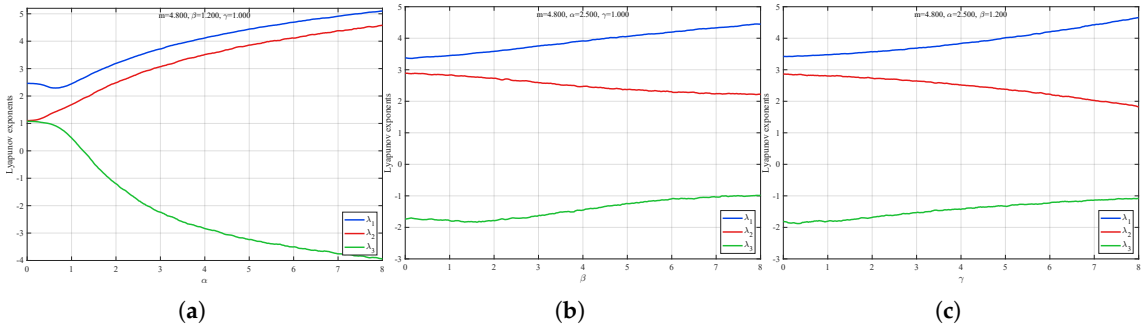
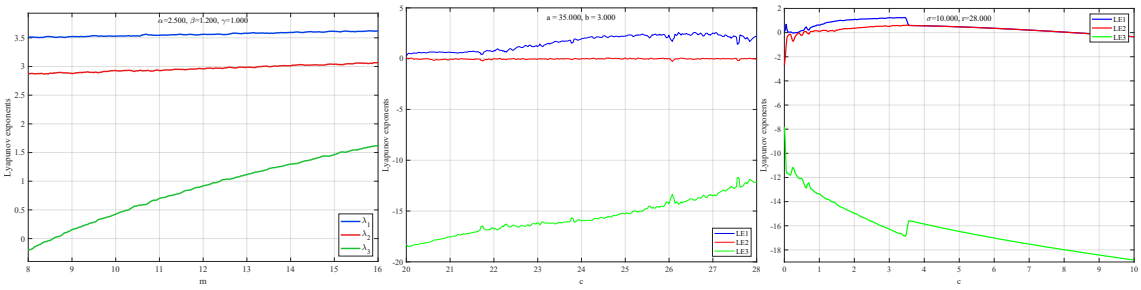


Figure 2. *Cont.*



(**d**) (**e**) (**f**)

Figure 2. Lyapunov exponents of different chaotic systems. (**a**) 3D-SBFCM system with varying α; (**b**) 3D-SBFCM system with varying β; (**c**) 3D-SBFCM system with varying γ; (**d**) 3D-SBFCM system with varying m; (**e**) Chen system with varying c; and (**f**) Lorenz system with varying c.

2.3.3. Permutation Entropy

Permutation entropy (PE) characterizes the complexity of a time series through the probability distribution of local ordering patterns. A larger PE value indicates greater randomness and temporal complexity in the sequence. The mathematical expression of PE is given in Equation (3).

$$PE = -\frac{\sum_{l=1}^{K} p_l \ln p_l}{\ln(m!)} \tag{3}$$

where m is the embedding dimension, $K = m!$ denotes the total number of possible permutation patterns, and p_l is the probability of occurrence of the l-th permutation pattern. In this paper, we set the embedding dimension to $m = 4$ and the delay to 1, and performed permutation entropy analysis on the 3D-SBFCM, 3D-ICPCM [32], 3D-BNM [33], Chen system, and Lorenz system. The results are shown in Figure 3.

Figure 3 shows that the 3D-SBFCM maintains relatively high and stable PE values over a wide range of parameter values, indicating strong temporal complexity in its output sequences. In contrast, 3D-ICPCM and 3D-BNM exhibit more obvious fluctuations in certain parameter intervals, which indicates that their time-series complexity is less stable under parameter variations. The Chen system exhibits a moderate PE level, whereas the Lorenz system shows relatively low PE values in certain parameter intervals, indicating relatively limited temporal complexity in these systems.

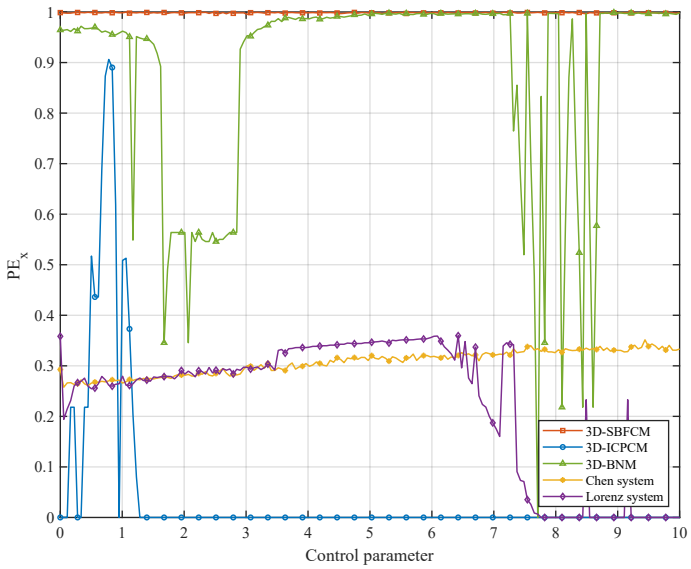


Figure 3. Permutation entropy of different chaotic systems.

2.3.4. 0-1 Test

The 0-1 test is used to determine whether a system exhibits chaotic behavior by constructing translation variables and examining the asymptotic growth of the mean square displacement. Its formulation is given in Equation (4).

$$\begin{cases} p(n) = \sum_{i=1}^{n} \phi(i)\cos(ic) \\ q(n) = \sum_{i=1}^{n} \phi(i)\sin(ic) \\ M(n) = \frac{1}{(N-n)} \sum_{i=1}^{N-n} \{[p(i+n) - p(i)]^2 + [q(i+n) - q(i)]^2\} \\ K_c = \mathrm{Corr}(\lambda, a) = \frac{\mathrm{Cov}(\lambda,a)}{\sqrt{\mathrm{Var}(\lambda)\,\mathrm{Var}(a)}} \\ \lambda = (1, 2, \ldots, n_{cut}) \\ a = (M(1), M(2), \ldots, M(n_{cut})) \\ K = \mathrm{median}(K_{c_1}, K_{c_2}, \ldots, K_{c_n}) \end{cases} \tag{4}$$

Here, $c \in (0, \pi)$ is a randomly selected parameter, and the statistic K is taken as the median of the results corresponding to multiple random values of c. When $K \approx 1$, the system is considered chaotic; when $K \approx 0$, the system is considered non-chaotic [34].

Figure 4 presents the 0-1 test results for the 3D-SBFCM, 3D-ICPCM, and 3D-BNM. It is clear that the K values corresponding to the three state variables of the 3D-SBFCM remain close to 1 throughout the tested parameter range, indicating that the proposed system can maintain stable chaotic behavior over a broad parameter interval.

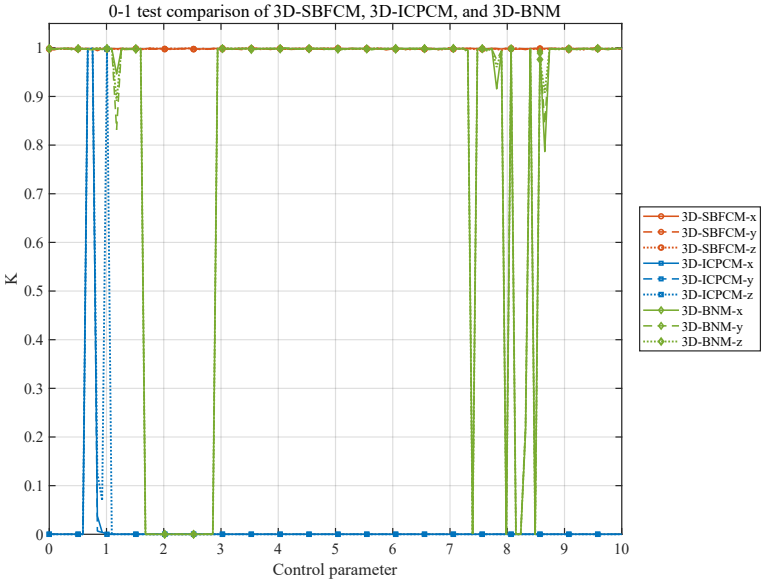


Figure 4. The 0-1 test results for different chaotic systems.

2.3.5. NIST SP 800-22 Test

To further verify the pseudorandomness of the output sequence generated by the 3D-SBFCM, the NIST SP 800-22 test suite [35] is employed for evaluation. For the test, the initial states were set to $x_0 = 0.1234$, $y_0 = 0.2345$, and $z_0 = 0.3456$. The results are listed in Table 1. The NIST SP 800-22 suite contains 15 test categories. Among them, the Random Excursion Test and the Random Excursion Variant Test include multiple state-dependent subtests. For compact presentation, Table 1 reports the minimum p-value among all valid state-dependent subtests for these two categories. All reported p-values are greater than 0.01, and all corresponding outcomes are marked as "Pass". These results demonstrate that the proposed 3D-SBFCM is capable of generating sequences with good statistical randomness, making it suitable for image encryption applications.

Table 1. NIST test values for 3D-SBFCM (P: Pass; F: Fail).

Test Item	p Value (x)	p Value (y)	p Value (z)	Results
Frequency	0.7733	0.1929	0.3513	Pass
Block Frequency	0.5785	0.1377	0.7387	Pass
Runs	0.5208	0.8731	0.5358	Pass
Longest Run of Ones	0.2942	0.8238	0.9236	Pass
Binary Matrix Rank	0.2221	0.2163	0.6639	Pass
Discrete Fourier Transform	0.8803	0.3127	0.9141	Pass
Non-Overlapping Template	0.5068	0.5032	0.4993	Pass
Overlapping Matching	0.9661	0.9969	0.9937	Pass
Universal Statistical	0.7717	0.8924	0.2446	Pass
Linear Complexity	0.2582	0.9793	0.1456	Pass
Serial	0.1655	0.2134	0.0982	Pass
Approximate Entropy	0.9172	0.0465	0.8234	Pass
Cumulative Sums	0.4185	0.1014	0.3286	Pass
Random Excursion	0.5638	0.2365	0.6063	Pass
Random Excursion Variant	0.0397	0.3244	0.3753	Pass

3. Encryption and Decryption Procedures

The purpose of image encryption is to weaken the spatial correlation and statistical regularity of plaintext images while maintaining strict reversibility. In block-based encryption algorithms, the partitioning strategy is crucial for confusion performance and for reducing residual structural information in the ciphertext. Although fixed-size partitioning is simple to implement, its regular block boundaries may retain certain spatial structures, thereby increasing the risk of visual information leakage. To address this issue, a color image encryption algorithm based on the proposed 3D-SBFCM is developed. The algorithm employs a hierarchical framework that incorporates plaintext-related key modulation, dynamic rectangular partitioning, inter-block permutation, intra-block permutation, dynamic S-box substitution, and forward–backward cross-channel diffusion. The overall framework of the proposed algorithm is illustrated in Figure 5.

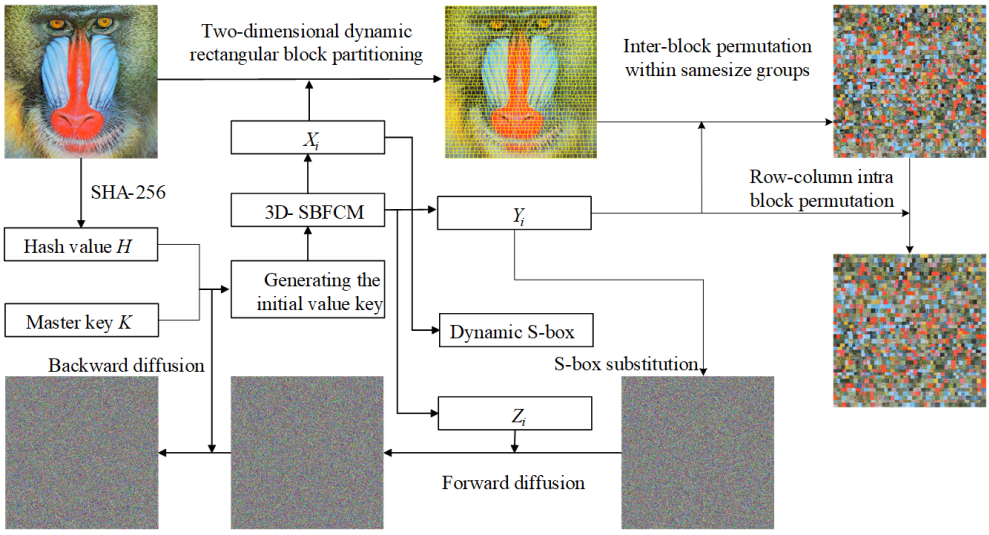


Figure 5. Flowchart of the proposed color image encryption algorithm.

3.1. Key Generation and Chaotic Sequence Construction

3.1.1. Master Key

The master key is defined as

$$K = \{x_0, y_0, z_0, m, \alpha, \beta, \gamma, \eta_1, \eta_2, \eta_3\}, \quad (5)$$

where (x_0, y_0, z_0) denote the initial states of the 3D-SBFCM; m, α, β, and γ denote the control parameters; and η_1, η_2, and $\eta_3 \in [0, 255]$ are auxiliary seeds.

3.1.2. Plaintext-Related Initialization

Let the input color image be denoted by

$$I \in \mathbb{Z}_{256}^{M \times N \times 3}, \tag{6}$$

where M and N denote the image dimensions, and $\mathbb{Z}_{256}$ denotes the gray-value domain $[0, 255]$.

The image is serialized in row-major order with the channel order R-G-B, and a SHA-256 digest is computed from the resulting byte stream. The digest is denoted by

$$H = \{h_1, h_2, \ldots, h_{32}\}, h_i \in [0, 255]. \tag{7}$$

To enable the receiver to regenerate the same chaotic control sequences during decryption, the digest H is transmitted together with the ciphertext as public session synchronization information. Since H is not secret, it is not regarded as part of the secret key and is not included in the key-space calculation.

Although an attacker may obtain or modify H, the valid chaotic sequences and diffusion feedback values are generated jointly from H and the secret master key. Therefore, modifying H can only affect the sequences generated at the receiver side, but cannot enable the attacker to predict, control, or reconstruct the valid encryption sequences without the secret master key. If a modified H is used in decryption, the generated sequences will be inconsistent with those used during encryption, resulting in decryption failure. Thus, such manipulation mainly leads to denial of decryption rather than plaintext recovery, key recovery, or a compromise of confidentiality.

The digest H and the auxiliary seeds η_1, η_2, and η_3 are further used to modulate the initial states of the 3D-SBFCM, producing the updated states (x_0', y_0', z_0'):

$$\begin{cases} x_0' = \mathrm{mod}\left(x_0 + \dfrac{\left[\left(\sum_{i=1}^{10} h_i\right) \bmod 256\right] \oplus \eta_1}{256}, 1 \right) \\ y_0' = \mathrm{mod}\left(y_0 + \dfrac{\left[\left(\sum_{i=11}^{21} h_i\right) \bmod 256\right] \oplus \eta_2}{256}, 1 \right) \\ z_0' = \mathrm{mod}\left(z_0 + \dfrac{\left[\left(\sum_{i=22}^{32} h_i\right) \bmod 256\right] \oplus \eta_3}{256}, 1 \right) \end{cases} \tag{8}$$

The 32-byte digest is partitioned into three segments of lengths 10, 11, and 11 for state modulation. In this way, the master key is coupled with the plaintext image, making the generated control sequences highly sensitive to plaintext variations.

The 3D-SBFCM is then iterated using (x_0', y_0', z_0') and the parameters m, α, β, and γ. After discarding the first N_0 transient values, three effective chaotic sequences are obtained:

$$X = \{x_1, x_2, \ldots, x_T\}, \quad Y = \{y_1, y_2, \ldots, y_T\}, \quad Z = \{z_1, z_2, \ldots, z_T\}. \tag{9}$$

where T denotes the total sequence length required by all encryption stages.

For integer-domain operations, the continuous sequences are quantized as:

$$\begin{cases} X_i = \mathrm{mod}\left(\left\lfloor x_i \times 10^{14} \right\rfloor, 256\right) \\ Y_i = \mathrm{mod}\left(\left\lfloor y_i \times 10^{14} \right\rfloor, 256\right) \\ Z_i = \mathrm{mod}\left(\left\lfloor z_i \times 10^{14} \right\rfloor, 256\right) \end{cases} \tag{10}$$

Specifically, the quantized sequence $\{X_i\}$ is used for dynamic partitioning and basic S-box construction, $\{Y_i\}$ is used for permutation control, and $\{Z_i\}$ is used for diffusion. To avoid reusing the same control variables in different modules, all quantized sequences are consumed sequentially by independent read pointers, which are advanced after each module is completed.

3.2. Two-Dimensional Dynamic Rectangular Partitioning

To reduce the structural regularity introduced by fixed-size blocking, a two-dimensional dynamic rectangular partitioning strategy is employed. In conventional fixed-size partitioning, block boundaries are located at regular row and column intervals, forming a periodic grid-like structure. Such a regular partitioning pattern may still leave residual boundary traces after block-level scrambling. In contrast, the proposed method first divides the image into horizontal strips with unequal heights and then partitions each strip into rectangular blocks with variable widths using chaotic sequences. Consequently, the boundary positions become non-periodic and dynamically controlled, which reduces the residual grid-like regularity caused by fixed block partitioning. Since only pixel positions are rearranged at this stage, reversibility is strictly preserved. For color images, identical block boundaries are used for the R, G, and B channels to maintain spatial consistency across channels.

To ensure that the remaining rows or columns can always be partitioned in a feasible manner, a residual feasibility condition is introduced. For a remaining length R, lower bound $d_{\min}$, and upper bound $d_{\max}$, the residual length is considered feasible if $R = 0$, or if there exists a positive integer n, representing the number of remaining segments, such that $nd_{\min} \le R \le nd_{\max}$.

Equivalently, the feasibility condition can be written as

$$\left\lceil \frac{R}{d_{\max}} \right\rceil \le \left\lfloor \frac{R}{d_{\min}} \right\rfloor. \tag{11}$$

This condition is used in the following dynamic height and width generation procedures to guarantee that the image can be completely partitioned without omission or overlap.

3.2.1. Dynamic Horizontal Strip Partitioning

Let $H_{\min}$ and $H_{\max}$ denote the minimum and maximum strip heights, respectively, satisfying

$$1 \le H_{\min} \le H_{\max} \le M. \tag{12}$$

The selected parameters should satisfy the feasibility condition for the whole image height, namely

$$\left\lceil \frac{M}{H_{\max}} \right\rceil \le \left\lfloor \frac{M}{H_{\min}} \right\rfloor. \tag{13}$$

Let $R_h^{(u)}$ denote the number of remaining unpartitioned rows after the first u strips have been determined. The initialization is given by

$$R_h^{(0)} = M. \tag{14}$$

If the image is finally divided into U strips and h_u denotes the height of the u-th strip, then

$$H_{\min} \le h_u \le H_{\max}, \quad u = 1, 2, \ldots, U. \tag{15}$$

and

$$\sum_{u=1}^{U} h_u = M. \tag{16}$$

The candidate strip height is first generated from the quantized chaotic sequence as

$$\hat{h}_u = H_{\min} + \left(X_{ptr_X} \bmod (H_{\max} - H_{\min} + 1)\right), \tag{17}$$

where ptr_X denotes the current reading pointer of $\{X_i\}$. After each generation, the pointer is updated as $ptr_X \leftarrow ptr_X + 1$. Here, $\hat{h}_u$ is only a preliminary candidate rather than the final height of the current strip. The actual strip height h_u is determined by checking whether the remaining number of rows can still be partitioned into valid strips.

To this end, we define a height feasibility predicate $\Phi_H(R)$. For a residual height R, $\Phi_H(R)$ returns true if R can be further divided into one or more valid strips whose heights lie within $[H_{\min}, H_{\max}]$. Specifically,

$$\Phi_H(R) = \text{true} \iff \left\lceil \frac{R}{H_{\max}} \right\rceil \le \left\lfloor \frac{R}{H_{\min}} \right\rfloor. \tag{18}$$

If the residual height after selecting $\hat{h}_u$ does not satisfy this condition, the candidate height is cyclically adjusted within the allowable interval until the residual height becomes feasible. The residual-constrained height determination procedure is summarized in Algorithm 1.

Algorithm 1 Residual-constrained determination of strip height.

Input: Residual height $R_h^{(u-1)}$, height bounds $H_{\min}$ and $H_{\max}$, candidate height $\hat{h}_u$
Output: Selected height h_u, updated residual height $R_h^{(u)}$

if $H_{\min} \le R_h^{(u-1)} \le H_{\max}$ **then**
  $h_u \leftarrow R_h^{(u-1)}$
else
  $h_u \leftarrow \hat{h}_u$
  $R_{\text{temp}} \leftarrow R_h^{(u-1)} - h_u$
  if $\Phi_H(R_{\text{temp}}) =$ false **then**
    $\Delta \leftarrow 1$
    while $\Delta \le H_{\max} - H_{\min}$ **do**
      $h_u \leftarrow H_{\min} + \left(\hat{h}_u - H_{\min} + \Delta\right) \bmod (H_{\max} - H_{\min} + 1)$
      $R_{\text{temp}} \leftarrow R_h^{(u-1)} - h_u$
      if $\Phi_H(R_{\text{temp}}) =$ true **then**
        break
      end if
      $\Delta \leftarrow \Delta + 1$
    end while
  end if
end if
$R_h^{(u)} \leftarrow R_h^{(u-1)} - h_u$
return h_u, $R_h^{(u)}$

After strip partitioning is completed, the starting and ending row indices of the u-th strip are given by

$$\begin{cases} r_u^{(s)} = 1 + \sum_{i=1}^{u-1} h_i \\ r_u^{(e)} = \sum_{i=1}^{u} h_i \end{cases} \tag{19}$$

Therefore, the u-th strip can be expressed as

$$S_u = I\left[r_u^{(s)} : r_u^{(e)}, 1 : N, :\right]. \tag{20}$$

3.2.2. Dynamic Column-Wise Partitioning Within Each Strip

After all dynamic strips are obtained, each strip is further partitioned along the column direction. Let $W_{\min}$ and $W_{\max}$ denote the minimum and maximum block widths, respectively, satisfying

$$1 \le W_{\min} \le W_{\max} \le N. \tag{21}$$

Similarly, the selected width parameters should satisfy

$$\left\lceil \frac{N}{W_{\max}} \right\rceil \le \left\lfloor \frac{N}{W_{\min}} \right\rfloor. \tag{22}$$

For the u-th strip S_u, let $R_{w,u}^{(v)}$ denote the number of remaining unpartitioned columns after the first v blocks have been determined. The initialization is

$$R_{w,u}^{(0)} = N. \tag{23}$$

Assume that the u-th strip is finally divided into T_u rectangular blocks, and let $w_{u,v}$ denote the width of the v-th block. Then,

$$W_{\min} \le w_{u,v} \le W_{\max}, \quad v = 1, 2, \ldots, T_u. \tag{24}$$

and

$$\sum_{v=1}^{T_u} w_{u,v} = N. \tag{25}$$

Within each strip, the subsequent control values of the quantized sequence $\{X_i\}$ are used to generate the candidate block width:

$$\hat{w}_{u,v} = W_{\min} + \left(X_{ptr_X} \bmod (W_{\max} - W_{\min} + 1)\right). \tag{26}$$

After each generation, the pointer is updated as $ptr_X \leftarrow ptr_X + 1$. Here, $\hat{w}_{u,v}$ is only a preliminary candidate rather than the final width of the current block. The actual block width $w_{u,v}$ is determined by checking whether the remaining number of columns can still be partitioned into valid blocks.

Similar to the strip-height determination, we define a width feasibility predicate $\Phi_W(R)$. For a residual width R, $\Phi_W(R)$ returns true if $R = 0$, or if R can be further divided into one or more valid blocks whose widths lie within $[W_{\min}, W_{\max}]$. Specifically,

$$\Phi_W(R) = \text{true} \iff \left\lceil \frac{R}{W_{\max}} \right\rceil \le \left\lfloor \frac{R}{W_{\min}} \right\rfloor. \tag{27}$$

If the residual width after selecting $\hat{w}_{u,v}$ does not satisfy this condition, the candidate width is cyclically adjusted within the allowable interval until the residual width becomes feasible. The residual-constrained width determination procedure is summarized in Algorithm 2.

Algorithm 2 Residual-constrained determination of block width.

Input: Residual width $R_{w,u}^{(v-1)}$, width bounds $W_{\min}$ and $W_{\max}$, candidate width $\hat{w}_{u,v}$
Output: Selected width $w_{u,v}$, updated residual width $R_{w,u}^{(v)}$

if $W_{\min} \le R_{w,u}^{(v-1)} \le W_{\max}$ **then**
  $w_{u,v} \leftarrow R_{w,u}^{(v-1)}$
else
  $w_{u,v} \leftarrow \hat{w}_{u,v}$
  $R_{\text{temp}} \leftarrow R_{w,u}^{(v-1)} - w_{u,v}$
  if $\Phi_W(R_{\text{temp}}) = \text{false}$ **then**
    $\Delta \leftarrow 1$
    while $\Delta \le W_{\max} - W_{\min}$ **do**
      $w_{u,v} \leftarrow W_{\min} + (\hat{w}_{u,v} - W_{\min} + \Delta) \bmod (W_{\max} - W_{\min} + 1)$
      $R_{\text{temp}} \leftarrow R_{w,u}^{(v-1)} - w_{u,v}$
      if $\Phi_W(R_{\text{temp}}) = \text{true}$ **then**
        break
      end if
      $\Delta \leftarrow \Delta + 1$
    end while
  end if
end if
$R_{w,u}^{(v)} \leftarrow R_{w,u}^{(v-1)} - w_{u,v}$
return $w_{u,v}$, $R_{w,u}^{(v)}$

After determining the block widths, the starting and ending column indices of the v-th block in the u-th strip are given by

$$\begin{cases} c_{u,v}^{(s)} = 1 + \sum_{j=1}^{v-1} w_{u,j} \\ c_{u,v}^{(e)} = \sum_{j=1}^{v} w_{u,j} \end{cases} \tag{28}$$

Accordingly, the v-th block in the u-th strip is defined as

$$B_{u,v} = I\left[r_u^{(s)} : r_u^{(e)},\, c_{u,v}^{(s)} : c_{u,v}^{(e)},\, :\right]. \tag{29}$$

Repeating the above procedure for all strips yields the complete set of two-dimensional dynamic rectangular blocks:

$$\mathcal{B} = \{B_{u,v} \,|\, 1 \le u \le U,\ 1 \le v \le T_u\}. \tag{30}$$

Let Q denote the total number of generated blocks. Then,

$$Q = \sum_{u=1}^{U} T_u. \tag{31}$$

The size of each block $B_{u,v}$ is

$$\text{size}(B_{u,v}) = h_u \times w_{u,v}. \tag{32}$$

According to the above construction, the image is partitioned into a set of rectangular blocks with irregular boundaries and variable sizes. These blocks are non-overlapping and collectively cover the entire image domain without omission. Therefore, the proposed partitioning scheme provides a reversible structural basis for the subsequent scrambling and reconstruction stages.

3.3. Inter-Block Permutation Based on Same-Size Grouping

After the two-dimensional dynamic rectangular partitioning, the image is decomposed into multiple rectangular blocks of different sizes. If blocks with different sizes are directly exchanged, boundary mismatch will occur during reconstruction. Therefore, a same-size grouping and intra-group permutation strategy is adopted for inter-block permutation.

Let the block set be denoted by

$$\mathcal{B} = \{B_1, B_2, \ldots, B_Q\}. \tag{33}$$

The size of the k-th block is denoted by

$$\text{size}(B_k) = (h_k, w_k), \tag{34}$$

where $\text{size}(\cdot)$ denotes the height–width pair of a block.

According to block size, all blocks are divided into P same-size groups:

$$\mathcal{G} = \{G_1, G_2, \ldots, G_P\}. \tag{35}$$

Assume that the p-th group contains n_p blocks. Then,

$$\begin{cases} G_p = \{B_{p,1}, B_{p,2}, \ldots, B_{p,n_p}\} \\ \sum_{p=1}^{P} n_p = Q \end{cases} \qquad p = 1, 2, \ldots, P \tag{36}$$

For each group G_p, a control vector of length n_p is extracted from the quantized chaotic sequence $\{Y_i\}$:

$$\mathbf{s}_p = \{s_{p,1}, s_{p,2}, \ldots, s_{p,n_p}\}. \tag{37}$$

The control vector $\mathbf{s}_p$ is sorted in ascending order to obtain the permutation index sequence.

$$\pi_p = \text{sortindex}(s_p). \tag{38}$$

The blocks within the p-th group are then rearranged according to π_p

$$B'_{p,i} = B_{p,\pi_p(i)}, \quad i = 1, 2, \ldots, n_p \tag{39}$$

If $n_p = 1$, the corresponding block remains unchanged in the inter-block permutation stage. Because this operation is carried out only among blocks with identical sizes, block boundary compatibility is preserved. Meanwhile, the global spatial layout of image regions is disrupted. During decryption, the original block positions can be exactly restored by applying the inverse permutation π_p^{-1}.

3.4. Two-Dimensional Intra-Block Permutation of Dynamic Rectangular Blocks

Although inter-block permutation changes the macroscopic positions of image regions, the local adjacency of pixels inside each block is still partially preserved. To further reduce local correlation, a two-dimensional intra-block permutation is performed within each block.

Let the k-th block after inter-block permutation be denoted by B'_k, whose size is $h_k \times w_k$. From the chaotic sequence $\{Y_i\}$, two control vectors are further extracted:

$$\begin{cases} a_k = a_{k,1}, a_{k,2}, \ldots, a_{k,h_k} \\ b_k = b_{k,1}, b_{k,2}, \ldots, b_{k,w_k} \end{cases} \tag{40}$$

By sorting a_k and b_k in ascending order, the row and column permutation sequences are obtained as

$$\begin{cases} \alpha_k = \text{sortindex}(a_k) \\ \beta_k = \text{sortindex}(b_k) \end{cases} \tag{41}$$

The rows of B'_k are first rearranged according to α_k, and the columns are then rearranged according to β_k. The resulting block after the intra-block permutation is denoted by

$$\widetilde{B}_k = \beta_k(\alpha_k(B'_k)). \tag{42}$$

The same row and column permutation sequences are shared by the R, G, and B channels so that inter-channel spatial consistency is preserved. After this stage, both the global positions of blocks and the local structures inside the blocks have been significantly disturbed, thereby preparing the image data for the subsequent value-domain encryption process.

An illustrative example of the inter-block and intra-block permutation process is shown in Figure 6.

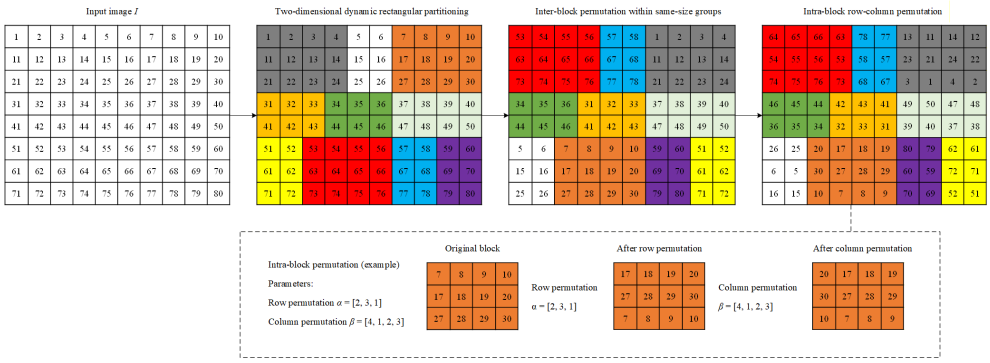


Figure 6. Example of inter-block and intra-block permutation based on dynamic rectangular partitioning.

3.5. Dynamic S-Box Substitution and Forward–Backward Cross-Channel Diffusion

The above scrambling operations effectively destroy the spatial correlation of the image, but the pixel intensities themselves remain largely unchanged. Therefore, a value-domain encryption stage is further introduced to enhance resistance against statistical and differential attacks. Specifically, dynamic S-box substitution is first employed to achieve nonlinear pixel transformation, and forward–backward cross-channel diffusion is then performed to propagate local perturbations over the entire image.

3.5.1. Dynamic S-Box Construction

Let the permuted color image obtained after the previous stages be denoted by

$$P = \{P_R, P_G, P_B\} \tag{43}$$

Using a unified scanning rule, the three channels are serialized into one-dimensional sequences of length $L = M \times N$:

$$\{p_R(t)\}_{t=1}^{L}, \{p_G(t)\}_{t=1}^{L}, \{p_B(t)\}_{t=1}^{L} \tag{44}$$

To construct the basic S-box, the first 256 mutually distinct integers are extracted from the quantized chaotic sequence $\{X_i\}$, forming the set

$$V_s = \{V_s(0), V_s(1), \ldots, V_s(255)\}. \tag{45}$$

If fewer than 256 distinct elements are obtained during the scanning process, the absent elements are supplemented in natural order until a complete permutation of $\{0, 1, \ldots, 255\}$ is formed. By sorting V_s in ascending order and recording the original indices, a permutation sequence is obtained:

$$\Pi = \{\pi(0), \pi(1), \ldots, \pi(255)\}. \tag{46}$$

Accordingly, the basic S-box is defined as

$$S_0(i) = \pi(i), \quad i \in \{0, 1, \ldots, 255\} \tag{47}$$

and its inverse mapping is denoted by S_0^{-1}.

To introduce position dependency, for channel $\chi \in \{R, G, B\}$ and pixel position t, the shift quantity $\mu_\chi(t)$ and mask quantity $v_\chi(t)$ are defined as

$$\begin{cases} \mu_\chi(t) = X_\chi(t) \bmod 256 \\ v_\chi(t) = Y_\chi(t) \bmod 256 \end{cases} \tag{48}$$

where $X_\chi(t)$ and $Y_\chi(t)$ denote the corresponding control mappings, and the next available control values are sequentially read from the quantized chaotic sequences $\{X_i\}$ and $\{Y_i\}$, respectively, for channel χ at position t.

Then, the dynamic S-box substitution result of channel χ at position t is given by

$$u_\chi(t) = S_0((p_\chi(t) + \mu_\chi(t)) \bmod 256) \oplus v_\chi(t) \tag{49}$$

After dynamic S-box substitution, three intermediate sequences are obtained:

$$\{u_R(t)\}_{t=1}^{L}, \{u_G(t)\}_{t=1}^{L}, \{u_B(t)\}_{t=1}^{L} \tag{50}$$

3.5.2. Forward Cross-Channel Diffusion

The keystreams required for forward and backward diffusion are all derived from the quantized chaotic sequence $\{Z_i\}$. To avoid reusing the same random segment in both diffusion stages, $\{Z_i\}$ is sequentially divided into six non-overlapping subsequences:

$$k_R^f(t) = Z_{6t-5}, \quad k_G^f(t) = Z_{6t-4}, \quad k_B^f(t) = Z_{6t-3} \tag{51}$$

$$k_R^b(t) = Z_{6t-2}, \quad k_G^b(t) = Z_{6t-1}, \quad k_B^b(t) = Z_{6t} \tag{52}$$

where Equation (51) corresponds to the forward diffusion keystreams, and Equation (52) corresponds to the backward diffusion keystreams.

To enhance the joint sensitivity of the diffusion process to both the master key and the plaintext content, the auxiliary seeds and the digest H are used to calculate the initial feedback vectors $q_R^{(0)}$, $q_G^{(0)}$, and $q_B^{(0)}$ of the forward diffusion, which are defined as

$$\begin{cases} q_R(0) = \eta_1 \oplus h_1 \oplus h_2 \oplus h_3 \oplus h_4 \\ q_G(0) = \eta_2 \oplus h_5 \oplus h_6 \oplus h_7 \oplus h_8 \\ q_B(0) = \eta_3 \oplus h_9 \oplus h_{10} \oplus h_{11} \oplus h_{12} \end{cases} \tag{53}$$

Let the intermediate cipher sequences after forward diffusion be denoted by

$$\{q_R(t)\}_{t=1}^{L}, \{q_G(t)\}_{t=1}^{L}, \{q_B(t)\}_{t=1}^{L} \tag{54}$$

Then, the forward cross-channel diffusion is defined as

$$\begin{cases} q_R(t) = u_R(t) \oplus k_R^f(t) \oplus q_R(t-1) \oplus q_B(t-1) \\ q_G(t) = u_G(t) \oplus k_G^f(t) \oplus q_G(t-1) \oplus q_R(t) \\ q_B(t) = u_B(t) \oplus k_B^f(t) \oplus q_B(t-1) \oplus q_G(t) \end{cases} \quad t = 1, 2, \ldots, L \tag{55}$$

This diffusion structure establishes progressive coupling among the three channels, thereby increasing the propagation range of a local perturbation across the whole image.

3.5.3. Backward Cross-Channel Diffusion

If only forward diffusion is employed, perturbations mainly propagate from the beginning to the end of the sequence, and the feedback from the latter part to the former part remains limited. To further enhance global coupling, a backward diffusion process from the end to the beginning is introduced.

Let $c_R(L+1)$, $c_G(L+1)$, and $c_B(L+1)$ denote the terminal feedback values of backward diffusion, defined as

$$\begin{cases} c_R(L+1) = \eta_1 \oplus h_{13} \oplus h_{14} \oplus h_{15} \oplus h_{16} \\ c_G(L+1) = \eta_2 \oplus h_{17} \oplus h_{18} \oplus h_{19} \oplus h_{20} \\ c_B(L+1) = \eta_3 \oplus h_{21} \oplus h_{22} \oplus h_{23} \oplus h_{24} \end{cases} \tag{56}$$

Let the final cipher sequences be denoted by

$$\{c_R(t)\}_{t=1}^{L}, \{c_G(t)\}_{t=1}^{L}, \{c_B(t)\}_{t=1}^{L} \tag{57}$$

The backward diffusion is defined as

$$\begin{cases} c_B(t) = q_B(t) \oplus k_B^b(t) \oplus c_B(t+1) \oplus c_R(t+1) \\ c_G(t) = q_G(t) \oplus k_G^b(t) \oplus c_G(t+1) \oplus c_B(t) \\ c_R(t) = q_R(t) \oplus k_R^b(t) \oplus c_R(t+1) \oplus c_G(t) \end{cases} \quad t = L, L-1, \ldots, 1 \tag{58}$$

By introducing reverse-direction feedback, perturbations generated in the tail part of the sequence can also affect the front part, which further strengthens the global diffusion capability of the proposed algorithm.

3.6. Decryption Procedure

The decryption procedure is the exact inverse of the encryption procedure. Let the input cipher image be denoted by

$$C = \{C_R, C_G, C_B\} \tag{59}$$

Using the same scanning rule as that adopted in the encryption stage, the three cipher channels are serialized into one-dimensional sequences of length L:

$$\{c_R(t)\}_{t=1}^{L}, \{c_G(t)\}_{t=1}^{L}, \{c_B(t)\}_{t=1}^{L} \tag{60}$$

The backward diffusion defined in Equation (58) is first inverted, with the terminal feedback values initialized according to Equation (56), to recover the intermediate sequences after forward diffusion:

$$\begin{cases} q_B(t) = c_B(t) \oplus k_B^b(t) \oplus c_B(t+1) \oplus c_R(t+1) \\ q_G(t) = c_G(t) \oplus k_G^b(t) \oplus c_G(t+1) \oplus c_B(t) \\ q_R(t) = c_R(t) \oplus k_R^b(t) \oplus c_R(t+1) \oplus c_G(t) \end{cases} \quad t = L, L-1, \ldots, 1 \tag{61}$$

Subsequently, the forward cross-channel diffusion defined in Equation (55) is inverted, with the initial feedback values initialized according to Equation (53), to recover the sequences after dynamic S-box substitution:

$$\begin{cases} u_R(t) = q_R(t) \oplus k_R^f(t) \oplus q_R(t-1) \oplus q_B(t-1) \\ u_G(t) = q_G(t) \oplus k_G^f(t) \oplus q_G(t-1) \oplus q_R(t) \\ u_B(t) = q_B(t) \oplus k_B^f(t) \oplus q_B(t-1) \oplus q_G(t) \end{cases} \quad t = 1, 2, \ldots, L \tag{62}$$

Next, inverse dynamic S-box mapping is performed to recover the permuted pixel values:

$$p_\chi(t) = \left[S_0^{-1}(u_\chi(t) \oplus v_\chi(t)) - \mu_\chi(t) \right] \bmod 256 \tag{63}$$

The recovered permuted sequences are then reorganized into a block set according to the regenerated dynamic rectangular boundaries. Let α_k^{-1} and β_k^{-1} denote the inverse row and column permutations of the k-th block, respectively. Then, the inverse intra-block permutation is expressed as

$$B'_k = \alpha_k^{-1}\left(\beta_k^{-1}(\widetilde{B}_k)\right) \tag{64}$$

After that, the inverse permutation π_p^{-1} is applied to each same-size block group to restore the original block positions. Finally, the restored R, G, and B channels are merged according to the regenerated dynamic block boundaries to reconstruct the original plaintext image I.

4. Security Analysis and Performance Evaluation

Unless otherwise specified, all experiments were conducted using MATLAB R2024a on a 64-bit Windows 10 Professional platform equipped with an Intel(R) Core(TM) i5-14600KF CPU @ 3.50 GHz, 32.0 GB RAM, and an NVIDIA GeForce RTX 5070 GPU with 12 GB video memory. No GPU acceleration or parallel computing toolbox was used in the reported experiments. The real-valued chaotic computations were performed using IEEE-754 double-precision floating-point arithmetic.

This section presents a systematic evaluation of the proposed 3D-SBFCM-based color image encryption algorithm with dynamic rectangular partitioning, dynamic S-box substitution, and forward-backward cross-channel diffusion. The assessment focuses on its security and practical robustness, including its ability to conceal plaintext information, resist common attacks, and maintain reliable performance under interference. Detailed analyses and experimental results are provided in the following sections.

4.1. Key Space Analysis

The secret key of the proposed algorithm consists of seven real-valued secret parameters, namely x_0, y_0, z_0, m, α, β, and γ, together with three auxiliary 8-bit seed parameters η_1, η_2, and η_3. Assuming that each real-valued secret parameter has an effective range of approximately one and is represented with a precision of 10^{-14}, these seven parameters pro-

vide approximately 10^{98} possible combinations. In addition, the three 8-bit seed parameters contribute 256^3 additional possibilities. Therefore, the total key space can be estimated as $10^{98} \times 256^3 \approx 2^{350}$. Since this value is much larger than the commonly accepted threshold of 2^{100} for resisting brute-force attacks, the proposed algorithm provides a sufficiently large key space to resist brute-force attacks in practical image encryption applications.

It should be noted that the above key-space estimation depends on the numerical precision adopted in the implementation. Since the simulations in this work were performed using IEEE-754 double-precision floating-point arithmetic, the precision level of 10^{-14} is consistent with the distinguishable numerical resolution for the considered parameter ranges. However, finite-precision representation may narrow the effective key space in practical implementations, particularly when single-precision or fixed-point arithmetic is adopted. Therefore, the estimated key space of approximately 2^{350} should be regarded as a theoretical estimate under the adopted double-precision simulation environment, and the same numerical representation, rounding rule, and iteration procedure should be used in both encryption and decryption.

4.2. Key Sensitivity Analysis

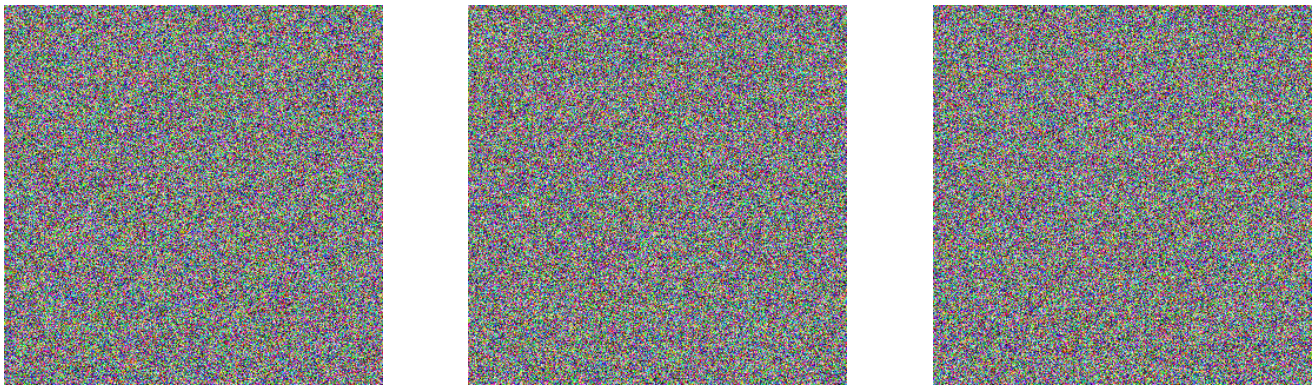
A desirable property of a secure encryption algorithm is high key sensitivity, such that even a negligible perturbation in the key produces completely different encryption or decryption outcomes. In the proposed algorithm, the permutation and diffusion operations are jointly governed by the initial conditions and control parameters of the chaotic system. Due to the inherent sensitivity of chaotic dynamics to initial conditions, a tiny perturbation in the key can cause substantial changes in the generated chaotic sequences, resulting in completely different ciphertexts and unsuccessful decryption.

To evaluate the key sensitivity of the proposed algorithm, the 512×512 Fruits image was selected for testing. Decryption was first carried out with the correct key and then repeated using four slightly modified keys generated by perturbing x_0, y_0, z_0, and m by 10^{-14}, respectively. The corresponding results are shown in Figure 7. It can be seen that the plaintext image can be correctly recovered only when the exact key is used, whereas even a negligible change in any of these parameters leads to decryption failure. These results confirm that the proposed algorithm possesses strong key sensitivity.



(**a**) (**b**) (**c**)

Figure 7. *Cont.*



(**d**) (**e**) (**f**)

Figure 7. Key sensitivity test results of the proposed algorithm for the Fruits image. (**a**) Plaintext image; (**b**) decrypted result using the correct key; (**c**) decrypted result with $x_0 + 10^{-14}$; (**d**) decrypted result with $y_0 + 10^{-14}$; (**e**) decrypted result with $z_0 - 10^{-14}$; and (**f**) decrypted result with $m - 10^{-14}$.

4.3. Histogram Analysis

Histogram analysis is commonly used to evaluate whether an encryption algorithm can effectively conceal the statistical characteristics of a plaintext image. Figure 8 shows the histograms of the House image before and after encryption for the three color channels. The histograms of the plaintext image are evidently nonuniform and contain several pronounced peaks, particularly in the high gray-level regions, indicating considerable statistical redundancy in the plaintext image. After encryption, however, the histograms of the R, G, and B channels become substantially flatter and more evenly distributed, and the dominant peaks disappear. This behavior indicates that the proposed algorithm can effectively conceal the statistical distribution of pixel values and significantly enhance the randomness of the encrypted image, which is beneficial for resisting statistical attacks.

4.4. Correlation Analysis

In plaintext images, adjacent pixels generally exhibit strong correlation in the horizontal, vertical, and diagonal directions. An effective image encryption algorithm should therefore disrupt this intrinsic dependency and render neighboring pixels statistically independent. To evaluate the decorrelation capability of the proposed method, the correlation coefficients of adjacent pixels in the Baboon image were calculated for the R, G, and B channels along the three aforementioned directions. The corresponding scatter plots and numerical results are provided in Figure 9 and Table 2, respectively. As can be observed from Figure 9a–c, the adjacent pixel pairs in the plaintext image are tightly clustered around the diagonal, reflecting the strong correlation naturally present in unencrypted images. By contrast, the scatter distributions of the encrypted image shown in Figure 9d–f are highly dispersed and exhibit no clear structural pattern, suggesting that the dependence between neighboring pixels has been effectively eliminated. This conclusion is further supported by the data in Table 2, where the correlation coefficients of the cipher image fall within the interval from -0.0014 to 0.0050, all of which are very close to zero. These results demonstrate that the proposed algorithm has a strong ability to eliminate adjacent-pixel correlation and thus provides effective protection against statistical analysis.

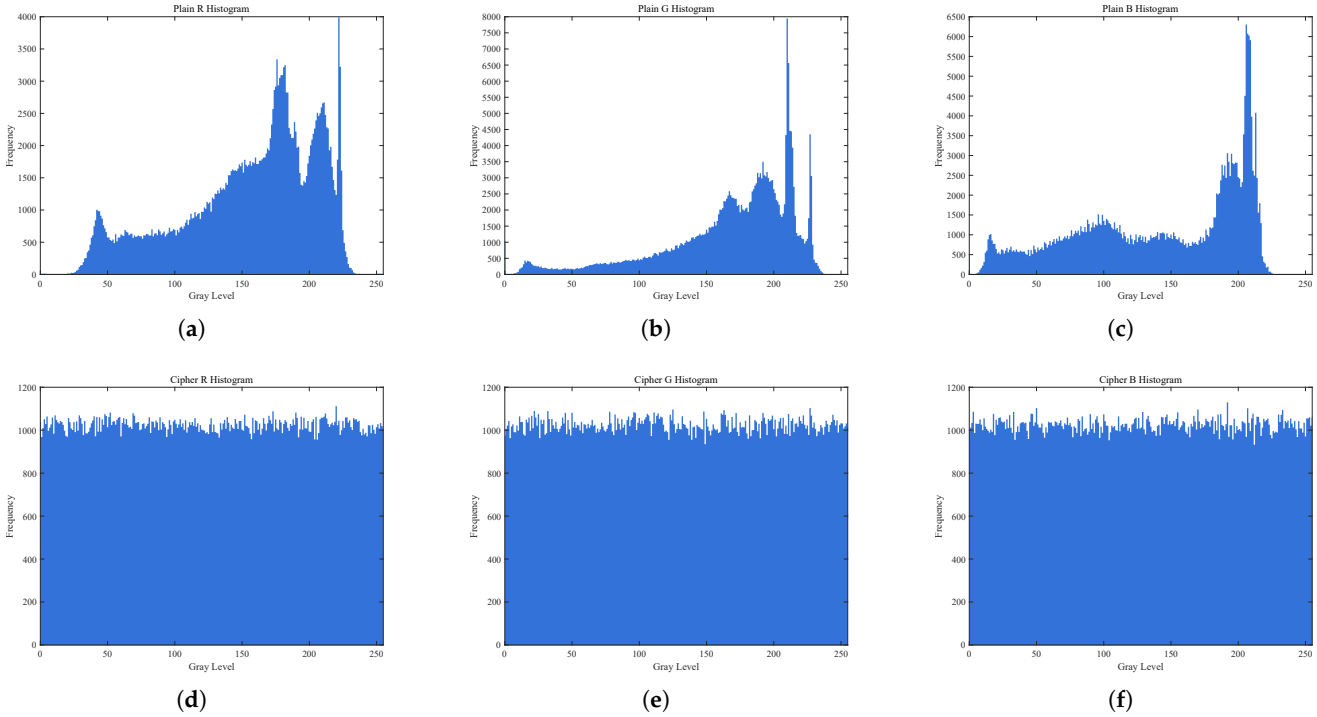


Figure 8. Histogram comparison of the RGB channels for the plaintext image and the corresponding cipher image: (**a**) plaintext-image R channel, (**b**) plaintext-image G channel, (**c**) plaintext-image B channel, (**d**) cipher image R channel, (**e**) cipher image G channel, and (**f**) cipher image B channel.

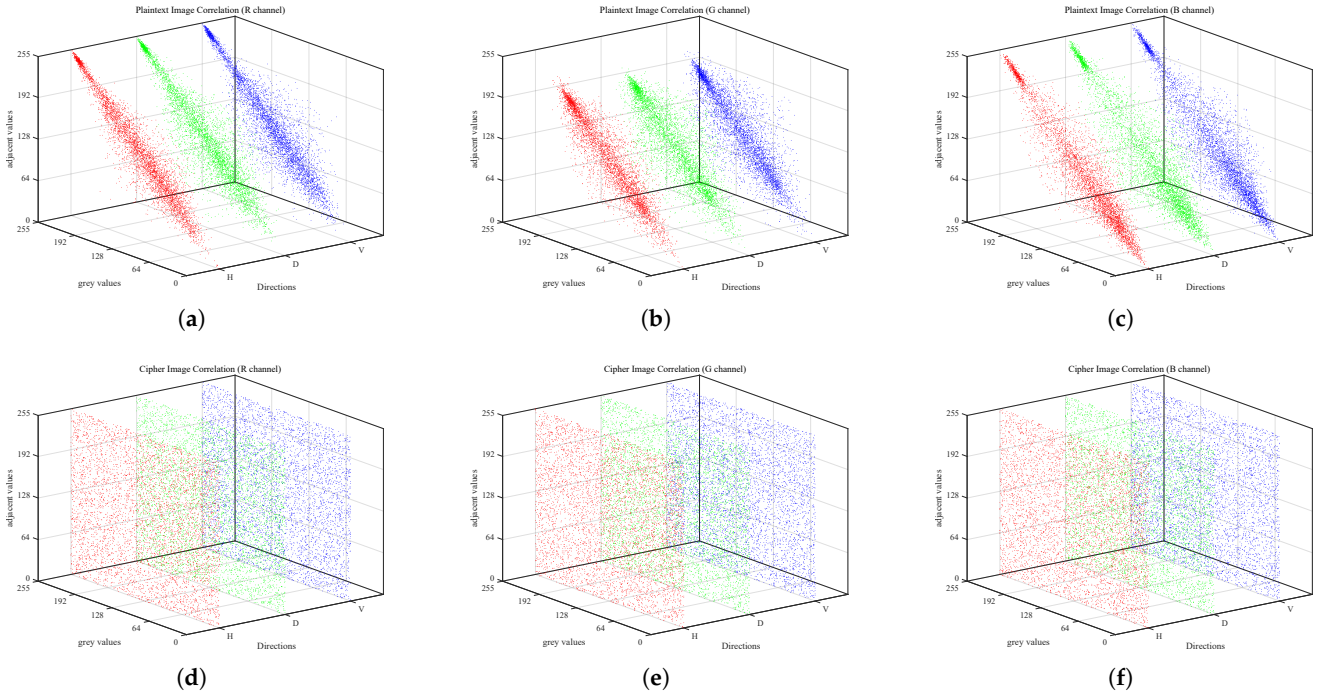


Figure 9. Correlation plots of the plaintext and cipher Baboon image. (**a**–**c**) correlation plots of the plaintext RGB components; (**d**–**f**) correlation plots of the cipher image RGB components.

Table 2. Comparison of correlation coefficients for the encrypted Baboon image.

Channels	Directions	Proposed Algorithm	Algorithm [36]	Algorithm [37]	Algorithm [38]
R	H	−0.0013	0.0085	0.0018	0.0009
	V	0.0050	−0.0025	−0.0050	−0.0028
	D	0.0007	−0.0151	−0.0096	0.0015
G	H	0.0016	0.0009	0.0043	0.0025
	V	0.0025	−0.0058	−0.0055	−0.0020
	D	−0.0014	−0.0007	0.0061	−0.0020
B	H	0.0020	0.0045	−0.0015	−0.0004
	V	0.0023	−0.0084	−0.0030	0.0010
	D	0.0009	−0.0048	−0.0029	−0.0028

4.5. Information Entropy Analysis

Information entropy is a standard metric for measuring the randomness of encrypted images. For an 8-bit image, the theoretical entropy of each channel is 8. A value closer to 8 indicates a more uniform pixel distribution and weaker statistical characteristics in the cipher image. It is given by

$$H(m) = -\sum_{i=0}^{255} P(m_i) \log_2 P(m_i) \tag{65}$$

where $P(m_i)$ denotes the probability of gray value m_i, and $\log_2(\cdot)$ is used for 8-bit entropy calculation. To evaluate the statistical security of the proposed algorithm, the Baboon image was selected for testing. The results are listed in Table 3.

Table 3. Information entropy of encrypted Baboon image.

Images	Proposed Algorithm	Algorithm [36]	Algorithm [37]	Algorithm [39]	Algorithm [40]
R channel	7.9993	7.9972	7.9993	7.9992	7.9915
G channel	7.9994	7.9975	7.9993	7.9993	7.9913
B channel	7.9993	7.9971	7.9993	7.9993	7.9914

As shown in Table 3, the proposed algorithm yields entropy values very close to the theoretical limit in all three channels. Its performance is comparable to, and in some cases better than, those of the reference algorithms. This indicates that the proposed algorithm can effectively eliminate the statistical structure of the plaintext image and generate cipher images with high randomness, thereby improving resistance to statistical attacks.

4.6. Differential Attack Analysis

Differential attack analysis is an important criterion for evaluating the sensitivity of an image encryption algorithm to slight changes in the plaintext. A robust encryption algorithm should ensure that even a one-pixel modification in the plaintext causes significant and widespread changes in the corresponding ciphertext. In this paper, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are employed for quantitative evaluation. Let C_1 and C_2 denote two $M \times N$ cipher images obtained by encrypting two plaintext images that differ only slightly. The corresponding metrics are defined as follows.

$$\begin{cases} NPCR = \dfrac{1}{M \times N} \displaystyle\sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \\ UACI = \dfrac{1}{M \times N} \displaystyle\sum_{i=1}^{M} \sum_{j=1}^{N} \dfrac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \end{cases} \tag{66}$$

For 8-bit images, the theoretical ideal values of NPCR and UACI are approximately 99.6094% and 33.4635%, respectively. The proposed algorithm strengthens nonlinear confusion through dynamic S-box substitution and accelerates disturbance propagation among RGB channels by means of a forward-backward cross-channel diffusion mechanism. As shown in Table 4, for the Baboon image, the NPCR values of the R, G, and B channels are 99.6063%, 99.6158%, and 99.6052%, respectively, while the corresponding UACI values are 33.4947%, 33.5159%, and 33.4486%. These results are very close to the theoretical values and are comparable to, and in some cases better than, those of the reference algorithms, indicating that the proposed algorithm exhibits high sensitivity to slight plaintext changes and strong resistance to differential attacks.

Table 4. NPCR and UACI results of the Baboon image for different encryption algorithms.

Images	NPCR			UACI		
	R	**G**	**B**	**R**	**G**	**B**
Proposed algorithm	99.6063	99.6158	99.6052	33.4947	33.5159	33.4486
Algorithm [39]	99.6197	99.6181	99.6250	33.4954	33.4996	33.5360
Algorithm [40]	99.6132	99.5995	99.6109	33.4543	33.4034	33.4275
Algorithm [41]	99.6114	99.6246	99.6195	33.4247	33.4585	33.4619

4.7. Robustness Analysis

In practical image transmission, cipher images may suffer from data loss, channel noise, or intentional tampering. Although an unauthorized attacker cannot recover the plaintext without the correct key, ciphertext degradation may still affect the quality of the decrypted results. Therefore, it is necessary to evaluate the robustness of the proposed algorithm under typical ciphertext degradation scenarios.

In this section, cropping attack, salt-and-pepper noise attack, and Gaussian noise attack are considered. The experimental procedure is as follows. First, the original image is encrypted to obtain the cipher image. Then, different attacks are applied to the cipher image. Finally, the attacked cipher image is decrypted using the correct key, and the recovered image is compared with the original image.

In this study, peak signal-to-noise ratio (PSNR) and bit correct rate (BCR) are adopted as the main quantitative metrics. Specifically, PSNR is mainly reported for cropping attacks to evaluate the visual reconstruction quality of the recovered images, whereas BCR is mainly reported for noise attacks to quantify the bit-level consistency between the recovered images and the original images. The mean square error, MSE, is defined as

$$MSE = \frac{1}{3MN} \sum_{\chi \in \{R,G,B\}} \sum_{i=1}^{M} \sum_{j=1}^{N} [I_\chi(i,j) - D_\chi(i,j)]^2 \tag{67}$$

where I and D denote the original image and the decrypted image from the attacked ciphertext, respectively. $M \times N$ is the image size, and χ represents the color channel.

The PSNR is defined as

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{68}$$

A higher PSNR value indicates better reconstruction quality.

The BER is defined as

$$BER = \frac{N_{\text{err}}}{N_{\text{total}}} \tag{69}$$

where N_{err} and N_{total} represent the number of erroneous bits and the total number of bits, respectively. Accordingly, BCR is given by

$$BCR = 1 - BER \tag{70}$$

A BCR value closer to 1 indicates higher bit-level consistency between the recovered image and the original image.

4.7.1. Cropping Attack

To evaluate the robustness against local ciphertext loss, four cropping patterns were applied to the cipher images, followed by decryption with the correct key. The visual and quantitative results are shown in Figure 10 and Table 5, respectively.

As shown in Figure 10, cropping attacks introduce different levels of distortion into the decrypted images. As the cropped area increases, the degradation becomes more evident. Nevertheless, the main visual content and structural contours of the recovered images remain recognizable even under severe cropping conditions.

The PSNR results in Table 5 show that the proposed algorithm exhibits a more gradual degradation trend than the compared algorithms as the cropping ratio increases. Although Algorithm [42] obtains a slightly higher PSNR under the 6.25% cropping attack, its performance drops sharply when the cropping ratio increases. In contrast, the proposed algorithm maintains competitive and relatively stable reconstruction quality under increasing cropping ratios. This demonstrates that the proposed algorithm has comparatively better robustness against ciphertext data loss.

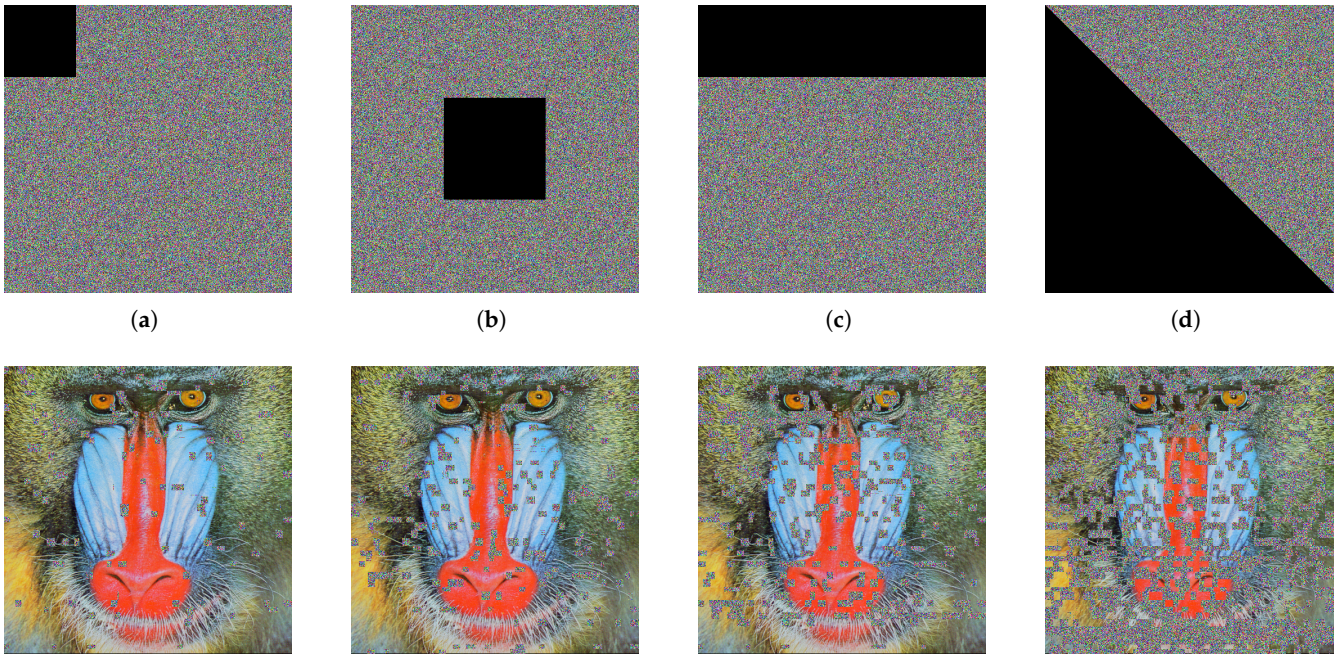


Figure 10. Decryption results under different cropping attacks: (**a**) upper-left cropping, 6.25%; (**b**) central cropping, 12.5%; (**c**) top-strip cropping, 25%; (**d**) diagonal triangular cropping, 50%; (**e**–**h**) the corresponding decrypted images.

Table 5. PSNR comparison under cropping attacks.

Algorithm	Cropping Attack (PSNR)			
	6.25%	**12.5%**	**25%**	**50%**
Proposed algorithm	20.7754	17.9212	15.6189	13.5742
Algorithm [42]	20.8054	15.6806	14.7826	7.7850
Algorithm [43]	20.6616	17.8724	15.2783	13.3415
Algorithm [44]	19.9545	17.3595	14.3256	12.2368

4.7.2. Salt-and-Pepper Noise Attack

To evaluate the robustness against impulse noise, salt-and-pepper noise with densities of 0.01, 0.02, 0.10, and 0.20 was added to the cipher images. The contaminated cipher images were then decrypted using the correct key. The visual results are shown in Figure 11.

At low noise densities, the recovered images preserve most visual details and exhibit only slight degradation. As the noise density increases, random black-and-white impulses cause more severe corruption in the decrypted images. However, the dominant image regions and basic structural information remain distinguishable. This suggests that the proposed algorithm can tolerate a certain degree of impulse noise in the ciphertext and still retain useful visual information.

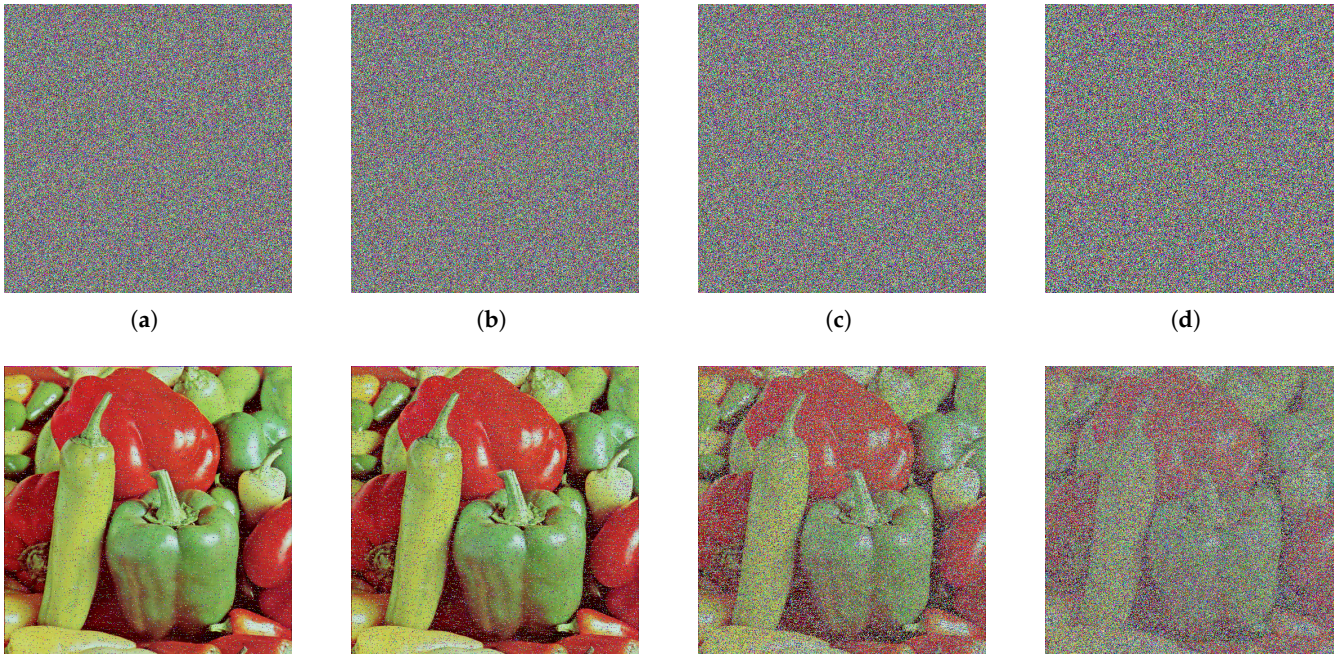


Figure 11. Decryption results of the proposed algorithm under salt-and-pepper noise attacks. (**a**–**d**) Cipher images contaminated with salt-and-pepper noise at densities of 0.01, 0.02, 0.1, and 0.2, respectively; (**e**–**h**) the corresponding decrypted images.

4.7.3. Gaussian Noise Attack

Gaussian noise with variances of 1×10^{-6}, 2×10^{-6}, 1×10^{-5}, and 2×10^{-5} was added to the cipher images to model channel noise contamination. Figure 12 illustrates the decrypted images obtained from the noisy ciphertexts.

Different from salt-and-pepper noise, Gaussian noise introduces continuous perturbations into the cipher images. As the noise variance increases, the recovered images become increasingly blurred and distorted. Even so, the principal scene information remains ob-

servable at the higher tested noise variances. This indicates that the proposed algorithm can still retain essential visual content under Gaussian channel noise interference.

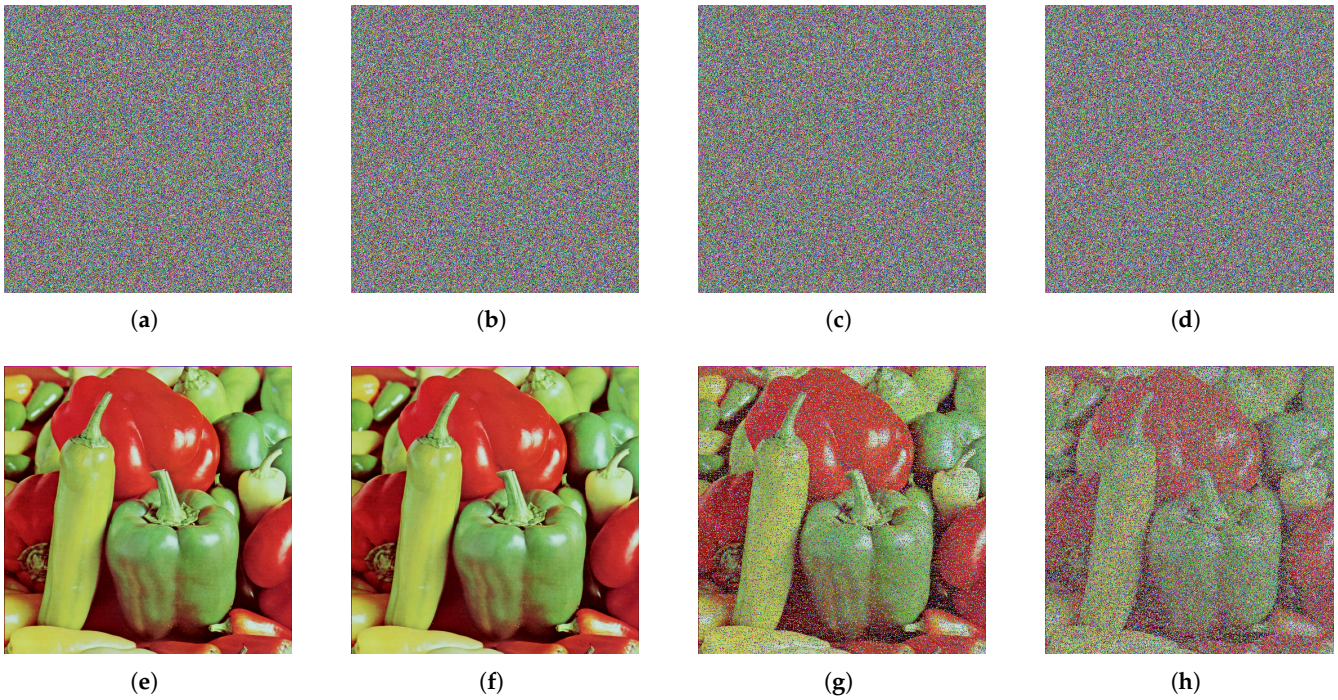


Figure 12. Decryption results of the proposed algorithm under Gaussian noise attacks. (**a**–**d**) Cipher images contaminated with Gaussian noise with variances of 1×10^{-6}, 2×10^{-6}, 1×10^{-5}, and 2×10^{-5}, respectively; (**e**–**h**) the corresponding decrypted images.

The quantitative BCR results for the Peppers image under the two noise attacks are summarized in Table 6. For RGB images, BCR is calculated separately for the red, green, and blue channels. Under salt-and-pepper noise, the BCR values decrease steadily as the noise density increases from 0.01 to 0.20. Specifically, the BCR values decrease from approximately 0.9958 to about 0.654. Under Gaussian noise, the BCR values remain nearly unchanged at low variances but drop noticeably when the variance increases to 1×10^{-5}, and 2×10^{-5}. These results confirm that stronger noise interference reduces bit-level reconstruction accuracy. Nevertheless, together with the visual results in Figures 11 and 12, the BCR values indicate that the proposed algorithm can still preserve useful visual information under both impulse noise and Gaussian channel noise.

Table 6. Channel-wise BCR robustness analysis of the Peppers image under noise attacks.

Noise Type	Noise Parameter	Red Channel	Green Channel	Blue Channel
SPN	0.01	0.9958	0.9956	0.9958
	0.02	0.9653	0.9647	0.9652
	0.10	0.7989	0.7988	0.7988
	0.20	0.6540	0.6546	0.6542
GN	1×10^{-6}	0.9999	0.9999	0.9999
	2×10^{-6}	0.9999	0.9999	0.9999
	1×10^{-5}	0.8526	0.8521	0.8526
	2×10^{-5}	0.6864	0.6809	0.6806

4.8. Efficiency Analysis

The efficiency of an image encryption algorithm is important for practical applications, especially in real-time transmission and resource-constrained scenarios. In this section,

the efficiency of the proposed algorithm is evaluated in terms of execution time and computational cost.

4.8.1. Execution Time Analysis

To evaluate the execution efficiency of the proposed algorithm, encryption and decryption experiments were conducted under the experimental environment described at the beginning of this chapter. The Lena color image with two different sizes, namely $256 \times 256 \times 3$ and $512 \times 512 \times 3$, was used for testing. Each experiment was independently repeated 20 times, and the average running time was recorded. The results are listed in Table 7.

Table 7. Execution time comparison of different encryption algorithms (Unit: s).

Image Size	Process	Proposed Algorithm	Algorithm [31]	Algorithm [45]
$256 \times 256 \times 3$	Encryption	0.9654	0.9925	2.0151
	Decryption	0.9347	0.9682	2.6891
$512 \times 512 \times 3$	Encryption	2.9875	4.0036	6.4257
	Decryption	2.6512	3.4552	9.4804

As shown in Table 7, for both Lena image sizes, the proposed algorithm requires less time for encryption and decryption than the compared methods.

4.8.2. Computational Complexity Analysis

Computational complexity is evaluated by considering both the estimated number of pixel-level operations and the asymptotic time complexity. For an RGB image of size $M \times N \times 3$, the proposed algorithm mainly includes chaotic sequence generation, dynamic rectangular partitioning, inter-block permutation, intra-block permutation, dynamic S-box substitution, and bidirectional diffusion.

The estimated costs of these operations are MN, MN, MN, MN, $3MN$, and $6MN$, respectively. Therefore, the total estimated computational cost is

$$MN + MN + MN + MN + 3MN + 6MN = 13MN \tag{71}$$

Since constant coefficients are ignored in asymptotic complexity analysis, the asymptotic time complexity of the proposed algorithm is $O(MN)$.

Table 8 compares the computational costs of different algorithms. Although the compared algorithms have the same asymptotic order of $O(MN)$, the proposed algorithm has a lower estimated computational cost in terms of the number of pixel-level operations.

Table 8. Computational complexity comparison of different encryption algorithms.

Algorithms	Estimated Computational Cost	Asymptotic Complexity
Proposed algorithm	$13MN$	$\mathcal{O}(MN)$
Algorithm [42]	$16MN$	$\mathcal{O}(MN)$
Algorithm [46]	$20MN$	$\mathcal{O}(MN)$
Algorithm [47]	$61MN + 3(M + N)$	$\mathcal{O}(MN)$

In terms of memory consumption, the proposed algorithm mainly requires storage for the input/output RGB images, image-scale intermediate arrays generated during permutation and diffusion, and chaotic control sequences used in different encryption stages. Since these data structures are all proportional to the image size, the memory requirement increases linearly with the number of pixels. Therefore, for an RGB image of size $M \times N \times 3$, the memory complexity of the proposed algorithm is $O(MN)$.

5. Conclusions

In this paper, a color image encryption method based on 3D-SBFCM, dynamic rectangular partitioning, and dynamic S-box substitution is proposed. The combination of chaotic dynamics, adaptive partitioning, and nonlinear substitution enhances the randomness and security of encrypted images. Experimental results verify the effectiveness and robustness of the proposed algorithm. In future research, the dynamic rectangular partitioning process will be further refined to enhance its adaptability to images with different textures and structures, while the dynamic S-box construction mechanism will be optimized to further strengthen the adaptability and security of the algorithm.

Author Contributions: T.W. and X.Y. designed the methodology; T.W. and C.X. implemented the encryption algorithm and simulation procedures; T.W., B.G. and H.W. performed the validation and security analysis; T.W. and X.Y. conducted the formal analysis; T.W., C.X. and H.W. carried out the investigation; T.W. and H.W. curated the data; T.W. and C.X. visualized the experimental results; T.W. wrote the original draft; X.Y., B.G., C.X. and H.W. reviewed and edited the manuscript; X.Y. and B.G. supervised the study; X.Y. administered the project; X.Y. and B.G. acquired the funding. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Open Fund of the State Key Laboratory of Digital Intelligent Technology for Unmanned Coal Mining, grant number SZQZ2025-2-003; the Natural Science Foundation of Anhui Province, grant number 2508085MF167; the National Natural Science Foundation of China, grant number 62102003; the Anhui Province New Era Education Quality Project (Graduate Education) Overseas Training, grant number 2024dshwyx019; the Anhui Province Young Key Teachers Overseas Study and Research Funding Project, grant number JWFX2023016; the Scientific Research Project of Higher Education Institutions of Anhui Province (Natural Science), grant number 2024AH051713; the Model Virtual Simulation Training Center for Information Security, grant number 2024sfxf013; and the Bengbu University 2024 General Program of University Research, grant number 2024ZR02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The standard test images used in this study are publicly available from commonly used image processing benchmark databases. The experimental results generated during this study are available from the corresponding author upon reasonable request.

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

References

1. Feng, L.; Du, J.; Fu, C.; Song, W. Image encryption algorithm combining chaotic image encryption and convolutional neural network. *Electronics* **2023**, *12*, 3455. [CrossRef]
2. Bodke, M.; Chaudhari, S. Enhanced selective hyper chaotic encryption using deep neural network for hyperspectral images. *Earth Sci. Inform.* **2025**, *18*, 161. [CrossRef]
3. Han, X.; Mou, J.; Jahanshahi, H.; Cao, Y.; Bu, F. A new set of hyperchaotic maps based on modulation and coupling. *Eur. Phys. J. Plus* **2022**, *137*, 523. [CrossRef]
4. Hu, H.; Cao, Y.; Hao, J.; Li, X.; Mou, J. A novel chaotic system with hidden attractor and its application in color image encryption. *Multimed. Tools Appl.* **2023**, *82*, 4343–4369. [CrossRef]
5. Zhou, Y.; Zhu, E.; Li, S. An image encryption algorithm based on the double time-delay Lorenz system. *Math. Biosci. Eng.* **2023**, *20*, 18491–18522. [CrossRef]
6. Inam, S.; Kanwal, S.; Niazi, M.; Karamti, H.; Al-Otaibi, S.; Jamjoom, M.M. A novel hybrid image encryption scheme using Henon map for secure image communication. *IET Inf. Secur.* **2026**, *2026*, 2676559. [CrossRef]
7. Pareek, N.K.; Patidar, V.; Sud, K.K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934. [CrossRef]
8. Aval, K.J.; Kamarposhty, M.S.; Damrudi, M. A simple method for image encryption using chaotic logistic map. *J. Comput. Sci. Comput. Math.* **2013**, *3*, 25–30. [CrossRef]

9. Belazi, A.; Abd El-Latif, A.A. A simple yet efficient S-box method based on chaotic sine map. *Optik* **2017**, *130*, 1438–1444. [CrossRef]
10. Jiang, N.; Dong, X.; Hu, H.; Ji, Z.; Zhang, W. Quantum image encryption based on Henon mapping. *Int. J. Theor. Phys.* **2019**, *58*, 979–991. [CrossRef]
11. Ratna, A.A.P.; Surya, F.T.; Husna, D.; Purnama, I.K.E.; Nurtanio, I.; Hidayati, A.N.; Purnomo, M.H.; Nugroho, S.M.S.; Rachmadi, R.F. Chaos-based image encryption using Arnold's cat map confusion and Henon map diffusion. *Adv. Sci. Technol. Eng. Syst.* **2021**, *6*, 316–326. [CrossRef]
12. Sun, S.; Yang, W.; Yin, Y.; Tian, X.; Li, G.; Deng, X. A color image encryption scheme utilizing a logistic-sine chaotic map and cellular automata. *Sci. Rep.* **2025**, *15*, 21603. [CrossRef] [PubMed]
13. Dua, S.; Kumar, A.; Dua, M.; Dhingra, D. ICFCM-MIE: Improved cosine fractional chaotic map based medical image encryption. *Multimed. Tools Appl.* **2024**, *83*, 52035–52060. [CrossRef]
14. Li, H.; Yu, S.; Feng, W.; Chen, Y.; Zhang, J.; Qin, Z.; Zhu, Z.; Wozniak, M. Exploiting dynamic vector-level operations and a 2D-enhanced logistic modular map for efficient chaotic image encryption. *Entropy* **2023**, *25*, 1147. [CrossRef]
15. Zhao, H.; Xie, S.; Zhang, J.; Wu, T. Fast image encryption algorithm based on improved Henon map. *Appl. Res. Comput.* **2020**, *37*, 3726–3730.
16. Sharma, M.; Chawla, P. An image encryption algorithm using a XOR enhanced 2D-Henon map. *Cluster Comput.* **2025**, *28*, 367. [CrossRef]
17. Zhang, X.; Zhang, H. A novel image encryption scheme using both pixel-level and bit-level permutations with an improved hyperchaotic system. *Cluster Comput.* **2025**, *28*, 388. [CrossRef]
18. Alexan, W.; Shabasy, N.H.E.; Ehab, N.; Maher, E.A. A secure and efficient image encryption scheme based on chaotic systems and nonlinear transformations. *Sci. Rep.* **2025**, *15*, 31246. [CrossRef]
19. Sun, C.; Wang, E.; Zhao, B. Image encryption scheme with compressed sensing based on a new six-dimensional non-degenerate discrete hyperchaotic system and plaintext-related scrambling. *Entropy* **2021**, *23*, 291. [CrossRef] [PubMed]
20. Kaur, M.; AlZubi, A.A.; Singh, D.; Kumar, V.; Lee, H.-N. Lightweight biomedical image encryption approach. *IEEE Access* **2023**, *11*, 74048–74057. [CrossRef]
21. Aljaedi, A.; Alharbi, A.R.; Aljuhni, A.; Alghuson, M.K.; Alassmi, S.; Shafique, A. A lightweight encryption algorithm for resource-constrained IoT devices using quantum and chaotic techniques with metaheuristic optimization. *Sci. Rep.* **2025**, *15*, 14050. [CrossRef] [PubMed]
22. Chen, X.; Mou, J.; Cao, Y.; Yan, H.; Jahanshahi, H. A chaotic color image encryption scheme based on improved Arnold scrambling and dynamic DNA encoding. *Multimed. Tools Appl.* **2023**, *82*, 43797–43818. [CrossRef]
23. Kumaran, V.N.S.; Manikandan, T.; Dhanaraj, R.K.; Al-Shehari, T.; Alsadhan, N.A.; Selvarajan, S. A secure medical image encryption technique based on DNA cryptography with elliptic curves. *Sci. Rep.* **2025**, *15*, 20003. [CrossRef]
24. Neela, K.L.; Kavitha, V. Blockchain based chaotic deep GAN encryption scheme for securing medical images in a cloud environment. *Appl. Intell.* **2023**, *53*, 4733–4747. [CrossRef]
25. Inam, S.; Kanwal, S.; Firdous, R.; Hajjej, F. Blockchain based medical image encryption using Arnold's cat map in a cloud environment. *Sci. Rep.* **2024**, *14*, 5678. [CrossRef]
26. Ahn, G.; Hong, S. QryptGen: A quantum GAN-based image encryption key generator using chaotic data distributions. *Quantum Inf. Process.* **2025**, *24*, 136. [CrossRef]
27. Nair, B.V.; Muni, S.S.; Durdu, A. Deep learning and chaos: A combined approach to image encryption and decryption. *arXiv* **2024**, arXiv:2406.16792. [CrossRef]
28. Li, C.; Zhang, Y.; Xie, E.Y. When an attacker meets a cipher-image in 2018: A year in review. *J. Inf. Secur. Appl.* **2019**, *48*, 102361. [CrossRef]
29. Gao, S.; Wu, R.; Iu, H.H.C.; Erkan, U.; Cao, Y.; Li, Q.; Toktas, A.; Mou, J. Chaos-based video encryption techniques: A review. *Comput. Sci. Rev.* **2025**, *58*, 100816. [CrossRef]
30. Ali, A.A. Chaotic performance for strange attractor and stability theory in dynamical systems. *Sch. J. Phys. Math. Stat.* **2026**, *3*, 107–117. [CrossRef]
31. Hu, J.; Zhang, J.; Ye, Y.; Jia, X.; Ma, M. Color image encryption scheme based on 2D chaotic mapping and circular shifting. *J. Supercomput.* **2025**, *81*, 1504. [CrossRef]
32. Verma, V.; Kumar, S.; Rani, N. Novel image encryption algorithm using hybrid 3D-ICPCM and hessenberg decomposition. *Nonlinear Dyn.* **2024**, *112*, 10583–10609. [CrossRef]
33. Verma, V.; Kumar, S. Quantum image encryption algorithm based on 3D-BNM chaotic map. *Nonlinear Dyn.* **2025**, *113*, 3829–3855. [CrossRef]
34. Su, Y.; Xia, H.; Chen, Z.; Chen, H.; Huang, L. A novel one-dimensional chaotic system for image encryption through the three-strand structure of DNA. *Entropy* **2025**, *27*, 776. [CrossRef] [PubMed]

35. Shi, R.H.; Yang, Y.G.; Xu, G.B.; Jiang, D.-H. Dynamic RGB image encryption algorithm with pixel-value-driven partitioning and a newly designed three-dimensional chaotic map. *Multimed. Syst.* **2026**, *32*, 85. [CrossRef]
36. Yu, J.; Peng, K.; Zhang, L.; Xie, W. Image encryption algorithm based on DNA network and hyperchaotic system. *Vis. Comput.* **2024**, *40*, 8001–8021. [CrossRef]
37. Yao, Y.; Xu, X.; Jiang, Z. A new chaotic color image encryption algorithm based on memristor model and random hybrid transforms. *Appl. Sci.* **2025**, *15*, 913. [CrossRef]
38. Cao, Y.; Song, Y. Color image encryption based on an evolutionary codebook and chaotic systems. *Entropy* **2024**, *26*, 597. [CrossRef]
39. Wang, S.; Pan, J.; Cui, Y.; Chen, Z.; Zhan, W. Fast color image encryption algorithm based on DNA coding and multi-chaotic systems. *Mathematics* **2024**, *12*, 3297. [CrossRef]
40. Xie, X.; Zhang, K.; Zheng, B.; Ning, H.; Zhou, Y.; Peng, Q.; Li, Z. A CML-ECA chaotic image encryption system based on multi-source perturbation mechanism and dynamic DNA encoding. *Symmetry* **2025**, *17*, 1042. [CrossRef]
41. Raghuvanshi, A.; Budhia, M.; Patro, K.A.K.; Acharya, B. FSR-SPD: An efficient chaotic multi-image encryption system based on flip-shift-rotate synchronous-permutation-diffusion operation. *Multimed. Tools Appl.* **2024**, *83*, 57011–57057. [CrossRef]
42. Qian, X.; Yang, Q.; Li, Q.; Liu, Q.; Wu, Y.; Wang, W. A novel color image encryption algorithm based on three-dimensional chaotic maps and reconstruction techniques. *IEEE Access* **2021**, *9*, 61334–61345. [CrossRef]
43. Tang, J.; Lu, M.; Zhang, Z.; Huang, X.; Huang, T.; Wang, J. Novel asymmetrical color image encryption using 2D sine-power coupling map. *Nonlinear Dyn.* **2024**, *112*, 11547–11569. [CrossRef]
44. Cun, Q.; Tong, X.; Wang, Z.; Zhang, M. A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing. *Vis. Comput.* **2023**, *39*, 6589–6608. [CrossRef]
45. Wang, W.; Sun, J.; Wang, G.; Zhang, H. Fisher-Yates scrambling algorithm combined with S-box color image encryption technology based on 3D-SCCM chaotic system. *Multimed. Tools Appl.* **2023**, *82*, 45233–45258. [CrossRef]
46. Mansoor, S.; Parah, S.A. HAIE: A hybrid adaptive image encryption algorithm using Chaos and DNA computing. *Multimed. Tools Appl.* **2023**, *82*, 28769–28796. [CrossRef]
47. Belazi, A.; Kharbech, S.; Aslam, M.N.; Talha, M.; Xiang, W.; Iliyasu, A.M.; El-Latif, A.A.A. Improved Sine-Tangent chaotic map with application in medical images encryption. *J. Inf. Secur. Appl.* **2022**, *66*, 103131. [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.