Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Color Image Encryption Based on Joint Parameter Optimization and Chaotic Key-Stream Reconstruction with Application to Medical Images

Dawei Wang

wdw0301@163.com

Shanxi Normal University

Wei Yu

Shanxi Normal University

Yajun Chen

Shanxi Normal University

Chenyang Li

Shanxi Normal University

Gaofeng Meng

Shanxi Normal University

Article

Keywords: Color image encryption, Joint parameter optimization, Chaotic system, Key-stream reconstruction, Plant Rhizome Growth Optimization, Medical image application

Posted Date: September 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10802384/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Color Image Encryption Based on Joint Parameter Optimization and Chaotic Key-Stream Reconstruction with Application to Medical Images

Dawei Wang [a,], Wei Yu [b], Yajun Chen [a], Chenyang Li [b], Gaofeng Meng [b]*
a School of Computer Science and Artificial Intelligence, Shanxi Normal University, Taiyuan 030031, China
b School of Physics and Electronic Engineering, Shanxi Normal University, Taiyuan 030031, China
** Corresponding author.*
E-mail address: wdw0301@163.com

Abstract

The widespread use of digital images in networked applications has raised increasing concerns regarding secure storage and reliable transmission of visual information. Nevertheless, current chaotic image encryption approaches generally focus on adjusting chaotic states or system parameters, whereas the variables associated with key-stream construction and subsequent encryption operations are rarely optimized in a unified manner. This separation may limit the adaptability between chaotic dynamics and the subsequent encryption process. In this paper, a secure color image encryption scheme based on joint parameter optimization and chaotic key-stream reconstruction is proposed. A unified control vector is constructed by integrating the parameters of chaotic sequence generation and key-stream reconstruction, and the optimal parameter combination is obtained through an improved Plant Rhizome Growth Optimization (IPRGO) algorithm. The optimized parameters are further applied to chaotic sequence generation, RGB synchronous permutation, and channel-coupled bidirectional nonlinear diffusion, thereby enhancing the interaction between parameter optimization and encryption operations. Experimental evaluations show that the generated ciphertext exhibits desirable statistical characteristics, reduced spatial dependency, and enhanced sensitivity to plaintext variations. Moreover, additional experiments on representative medical images verify the applicability of the proposed method for sensitive image protection scenarios. These results further suggest that the proposed framework can be extended to sensitive medical image protection scenarios while maintaining stable encryption performance.

Keywords: Color image encryption; Joint parameter optimization; Chaotic system; Key-stream reconstruction; Plant Rhizome Growth Optimization; Medical image application

1 Introduction

With the continuous development of digital information systems, images have become an important carrier for visual information transmission and storage in modern applications [1]. These visual data are widely applied in various fields, including industrial monitoring, remote sensing, medical diagnosis, and intelligent information services. Unlike textual information, images contain massive pixel data, significant redundancy, and strong spatial correlations among neighboring pixels[2]. Moreover, the correlation among red, green, and blue (RGB) channels introduces additional challenges for secure color image processing[3,4]. During transmission and storage, unauthorized access, modification, or leakage of image data may result in privacy disclosure and information security risks[5]. Therefore, developing secure and efficient color image encryption methods remains an important research issue in multimedia security.

To address the security requirements of visual data transmission, researchers have explored various encryption mechanisms for protecting image information in digital environments[6]. Traditional cryptographic primitives, including symmetric encryption algorithms, offer strong confidentiality guarantees for general digital data. Nevertheless, when directly applied to images, these methods mainly operate on binary representations and do not exploit the statistical and spatial characteristics naturally embedded in image data[7]. Such limitations become more evident because image data inherently contain considerable redundancy and non-negligible correlations between adjacent pixels. In this context, chaotic dynamics have been introduced into

image encryption owing to their complex evolution characteristics and capability of generating deterministic yet highly irregular sequences. The sensitivity of chaotic trajectories to system states and their nonlinear coupling properties enable chaotic sequences to be employed in permutation index generation, key-stream construction, and pixel diffusion processes[8–10]. Based on this mechanism, most chaos-driven encryption frameworks combine spatial permutation and value diffusion operations to disrupt pixel arrangements and weaken the statistical relationship between plaintext and ciphertext images[11,12].

Based on chaos-driven encryption architectures, recent studies have attempted to enhance color image security by improving chaotic system design, spatial permutation strategies, cross-channel interactions, and plaintext-dependent key mechanisms[13–15]. With the increasing demand for secure medical data transmission and cloud-based healthcare services, encryption techniques have also been introduced into medical image protection to preserve sensitive diagnostic information and patient privacy[16–18]. However, medical images usually contain complex visual structures and significant information redundancy, which require encryption schemes with strong randomness, effective diffusion capability, and reliable reversibility. Meanwhile, for general color image encryption, the coordination between key generation, key-stream construction, and encryption operations remains insufficiently explored. Therefore, developing an adaptive encryption framework that can effectively regulate both chaotic generation and subsequent encryption processes is still an important research direction.

Since the performance of chaotic encryption schemes is highly dependent on parameter configurations, optimization-based strategies have gradually been explored to obtain more suitable encryption parameters. Various metaheuristic approaches, including evolutionary and swarm-based algorithms, have been applied to optimize chaotic initial states, control coefficients, and encryption-related variables[19–21]. By adjusting these parameters, existing studies aim to enhance encryption performance from statistical and differential perspectives, including ciphertext randomness, pixel correlation reduction, and resistance against differential attacks. These investigations demonstrate the potential of optimization techniques in improving the adaptability of encryption parameter selection. However, most existing optimization-assisted encryption schemes mainly regard optimized variables as independent chaotic settings or external key parameters, while their influence on subsequent key-stream reconstruction, channel interaction, and diffusion processes remains insufficiently investigated.

Despite the improvements achieved by optimization-assisted chaotic encryption methods, several limitations remain in the current design framework. First, the optimization target is often limited to chaotic initial values or a few control parameters, whereas the variables that control key-stream rearrangement, channel coupling, and diffusion scheduling receive less attention. Second, the optimized parameters are generally regarded as external settings, and their potential influence on subsequent encryption operations, such as permutation generation and key-stream construction, has not been fully exploited. Third, the interaction mechanism among RGB channels in some existing schemes remains insufficient, which may restrict the propagation capability of plaintext variations across different color components[22–24]. These observations indicate that future color image encryption frameworks should establish a closer connection between parameter optimization and the complete encryption procedure, allowing optimized variables to jointly regulate chaotic generation, key-stream reconstruction, and channel-coupled permutation–diffusion operations.

To overcome the above limitations, this paper proposes a color image encryption framework driven by joint parameter optimization and chaotic key-stream reconstruction. Specifically, the control parameters of the four-dimensional Lorenz system and key-stream reconstruction variables are integrated into a unified optimization space, where the proposed improved plant rhizome growth optimization algorithm searches for adaptive parameter combinations. The obtained parameters are further incorporated into chaotic sequence generation, row–column permutation construction, RGB key-stream reconstruction, and channel-coupled bidirectional diffusion, enabling the optimization process to directly participate in encryption execution. The main

contributions of this paper are summarized as follows. First, a joint parameter optimization strategy is established by incorporating chaotic parameters and key-stream reconstruction variables into a unified control model, extending conventional parameter optimization approaches in image encryption. Second, a chaotic key-stream reconstruction mechanism is designed through segment rearrangement, circular shifting, and channel fusion to enhance the flexibility of encryption sequence construction. Third, an RGB synchronous permutation and channel-coupled bidirectional nonlinear diffusion structure is developed to improve spatial scrambling and inter-channel information interaction. Finally, an improved plant rhizome growth optimization algorithm incorporating adaptive growth regulation and stagnation-based local updating is introduced to improve optimization efficiency and search capability. Comprehensive experiments, including security analysis, comparative evaluation, and medical image application validation, are conducted to verify the effectiveness and applicability of the proposed scheme.

2 Fundamentals

2.1 Four-dimensional Lorenz system

Chaotic systems are suitable for image encryption because their iterative outputs can provide deterministic but difficult-to-predict numerical sequences. In an encryption procedure, such sequences can be converted into permutation indices or diffusion key streams. Compared with low-dimensional maps, a four-dimensional chaotic model introduces more state variables and a wider parameter space, which is helpful for producing richer sequence variations during pixel scrambling and diffusion.

In this work, the four-dimensional Lorenz system is used as the basic sequence generator. Its mathematical model is expressed as follows:

$$\begin{cases} \dot{x} = a\left(y - x\right) + w \\ \dot{y} = cx - xz - y \\ \dot{z} = xy - bz \\ \dot{w} = -yz + rw \end{cases} \tag{1}$$

where, x, y, z and w are the state variables, a, b, c and r are the control parameters. When $a = 10, b = 8/3, c = 28$ and $r = -1$,the system exhibits hyperchaotic behavior.

To examine the dynamic behavior of the selected system, $r = -1$ is used in the phase-space analysis, and the initial state is set as $(x_0, y_0, z_0, w_0) = (1,1,1,1)$ .The system is solved by the fourth-order Runge–Kutta method, and the corresponding phase portraits are shown in Fig.1, The trajectories do not converge to periodic or closed orbits; instead, they spread over different phase planes with evident stretching and folding. In particular, the distributions in the $x - z$ and $y - z$ planes show double-scroll characteristics, indicating complex state evolution. These dynamic features support the use of the generated sequences in the subsequent construction of row-column permutation indices and diffusion key streams.

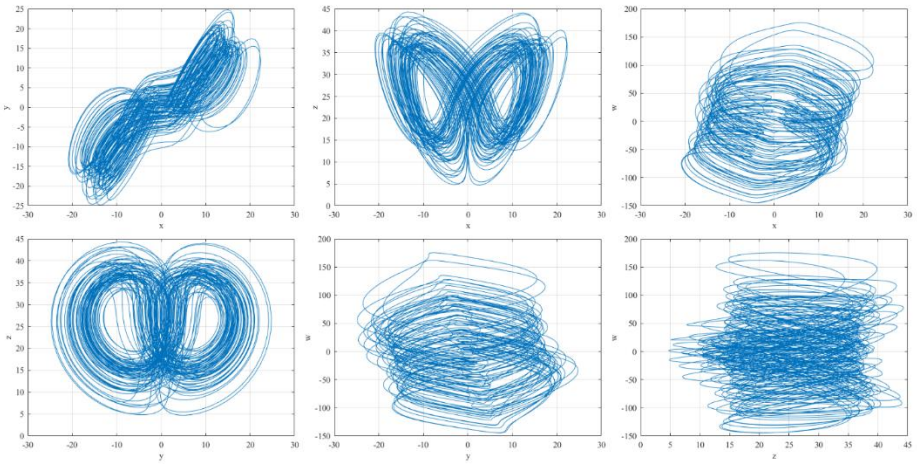


Fig.1 Phase portraits of the four-dimensional Lorenz system in different phase planes.

2.2 Basic principle of plant rhizome growth optimization

Plant Rhizome Growth Optimization (PRGO) is a population-based search algorithm inspired by the growth behavior of plant root systems. In PRGO, an individual corresponds to one candidate solution, and different root-growth patterns are modeled as different position-update operators. The taproot operator mainly generates candidate movements with larger search spans, whereas the fibrous-root operator introduces local adjustment and stochastic diffusion around existing individuals. Through the cooperation of these two operators, PRGO can perform global search and local refinement during the iterative process.

2.2.1 Taproot growth model

The taproot growth model is used to produce new candidate positions by combining several reference individuals, including the current best solution, the population mean, and randomly selected individuals. Instead of relying on a single update direction, this model generates candidate seeds through three complementary forms, namely deep taproot updating, mean-guided lateral-root updating, and best-guided fibrous-root updating.

1) Directional Growth of Fibrous Roots

$$Seed_1 = X_i^t + \alpha_1 \left(X_{best} - X_{r1}^t \right) \tag{2}$$

where $Seed_1$ denotes the candidate solution generated for individual i at iteration t. X_i^t is the current position of individual i. X_{best} represents the best solution found so far, and X_{r1}^t is a randomly selected individual from the current population. The coefficient α_1 determines the search tendency of this update. When $\alpha_1 \in (0,1.5)$, the candidate solution is guided by the difference between the best individual and a randomly selected individual, which strengthens exploitation around promising regions. When $\alpha_1 \in (-0.5,0)$, the sign of the update direction changes, introducing a reverse movement that helps preserve exploratory behavior. Thus, this rule combines best-solution guidance with random-direction perturbation to update candidate positions.

2) Mean-Guided Lateral-Root Growth

$$Seed_2 = X_{mean}^t + \alpha_2 \left(X_{mean}^t - X_{r2}^t \right) \tag{3}$$

where $Seed_2$ denotes the candidate solution generated by the lateral-root update at iteration t. X_{mean}^t is the mean position of the current population, and X_{r2}^t represents an individual randomly selected from the population. The coefficient α_2 controls the direction and magnitude of the movement around the population mean. When $\alpha_2 \in (-1,0)$, the generated candidate moves in one perturbation direction relative to the mean position, when $\alpha_2 \in (0,1)$, the movement direction is reversed. Through this mean-guided update, the algorithm can generate new candidate positions around the population center while avoiding an excessively concentrated search in a

single local region.

3) Deep Nutrient Absorption of Taproot

$$Seed_3 = Seed_2 + \alpha_3 \left(Seed_1 - R_l^u \right) \tag{4}$$

where $Seed_3$ denotes the candidate solution obtained from the taproot updating rule. R_l^u is a random position within the feasible search interval, l and u are the lower and upper bounds of the search space, respectively. The coefficient α_3 is calculated as $\alpha_3 = \varepsilon \times rand(0,1) + \tilde{\varepsilon}, \tilde{\varepsilon} = 1 - \varepsilon$, where ε is a binary random variable. When $\varepsilon = 1$, α_3 takes a random value in $(0,1)$; when $\varepsilon = 0$, $\alpha_3 = 1$. Thus, the update rule can switch between stochastic perturbation and full-step movement. The difference term $Seed_1 - R_l^u$ provides directional information for updating $Seed_2$ and generating $Seed_3$.

2.2.2 Fibrous-root growth model

The fibrous-root growth model provides another update operator for PRGO. Unlike the taproot model, this operator emphasizes dispersed movements generated from the current individual and randomly selected individuals. It is therefore used to introduce additional candidate directions during the search process and prevent the population from being restricted to a narrow region. The update rule is given as follows:

$$\begin{aligned} X_c &= \frac{\alpha_4 \left(X_{r3} + X_{r4} + X_i^t \right)}{3} \\ Seed_4 &= X_c + \left(X_{lbest} - X_i^t + X_{r3} + X_{r4} \right) \end{aligned} \tag{5}$$

where X_c is an intermediate candidate position generated by the fibrous-root diffusion process, and $Seed_4$ denotes the candidate solution obtained from this update rule. The coefficient α_4 is defined as $\alpha_4 = \sigma \times rand(0,1)$, where σ is a binary random variable. When $\sigma = 1$, $\alpha_4 \in (0,1)$, and the diffusion component X_c is activated. When $\sigma = 0$, $\alpha_4 = 0$, and the diffusion component is suppressed. X_{lbest} represents the local best solution used to guide the fibrous-root update, while X_{r3} and X_{r4} are two randomly selected individuals from the population. Here, r_3 and r_4 are distinct random integers selected from $\{1, 2, ..., N\}$. In this way, the fibrous-root update can generate additional search directions through randomly selected individuals, which helps maintain the diversity of candidate solutions during the iterative search.

3 Improved plant rhizome growth optimization algorithm

3.1 Overall framework of the proposed IPRGO algorithm

In the proposed encryption scheme, parameter optimization is not an isolated numerical search task. Each candidate solution must be evaluated through chaotic sequence generation, key-stream reconstruction, permutation-diffusion encryption, and security-metric calculation. Therefore, the search process is constrained by a limited number of function evaluations and requires stable coordination between wide-range exploration and local refinement. Although the original PRGO provides two different update modes through taproot and fibrous-root growth, its mode selection is usually controlled by fixed or empirically assigned parameters. Such a setting may not match the search requirements at different optimization stages. If the early search is insufficiently dispersed, the population may converge to a limited region too quickly; if large-step movements remain frequent in the later stage, the convergence trend may be disturbed. In addition, when the best fitness value remains unchanged for several consecutive iterations, the original PRGO does not provide a dedicated strategy to introduce new candidate directions without discarding useful search information.

To address these issues, this paper develops an improved plant rhizome growth optimization algorithm, denoted as IPRGO. Two mechanisms are incorporated into the original PRGO framework. First, an adaptive growth-mode probability adjustment mechanism is used to regulate the activation probability of different growth modes according to the optimization progress. Second, a stagnation-detection-driven local reset

mechanism is introduced to reinitialize a subset of inferior individuals when the search becomes stagnant, while retaining high-quality individuals. The overall procedure of IPRGO is shown in Fig.2. After population initialization, candidate solutions are evaluated by the encryption-oriented fitness function. Then, the growth-mode probability is updated, the population has evolved through PRGO-based position updating, and stagnation detection is performed. If the stagnation condition is satisfied, local reset is applied to selected individuals. Finally, boundary correction, fitness re-evaluation, and global-best updating are conducted iteratively until the stopping condition is reached.

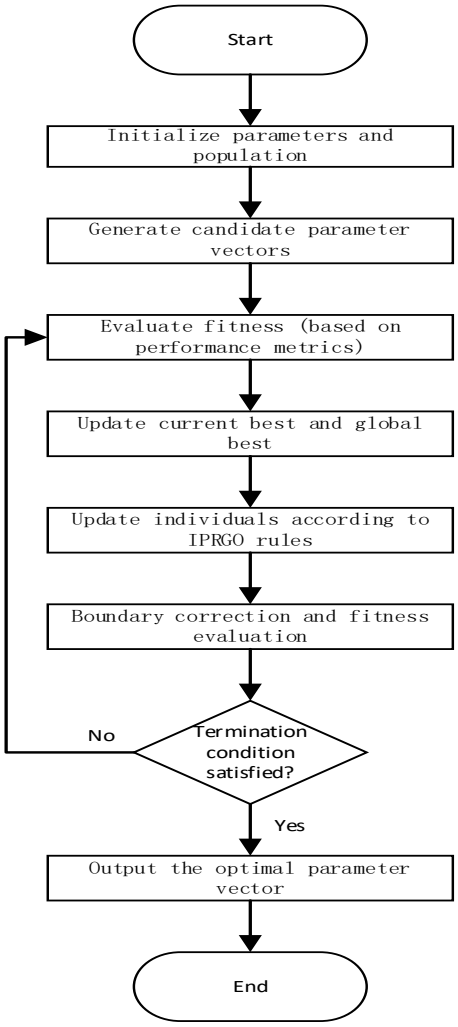


Fig.2 Flowchart of the proposed IPRGO algorithm

3.2 Adaptive growth-mode probability adjustment mechanism

In the original PRGO, the selection between taproot growth and fibrous-root growth is usually determined by a fixed probability. However, in the encryption-parameter optimization task considered in this paper, the search requirement varies with the optimization progress. At the beginning, candidate solutions should be distributed over a wider region to explore different combinations of chaotic-system parameters and perturbation-layer variables. In the later stage, the search should gradually focus on local refinement around high-quality parameter vectors. For this reason, IPRGO introduces a progress-dependent growth-mode probability. Let FEs denote the current number of function evaluations, and let $MaxFEs$ denote the maximum number of function evaluations. The normalized search progress is defined as：

$$t = \frac{FEs}{MaxFEs}$$
$$p_{tap}(t) = p_{max} - (p_{max} - p_{min}) \cdot t \tag{6}$$

where $t \in [0,1]$, $p_{\max}$ and $p_{\min}$ are the upper and lower bounds of the taproot activation probability, respectively. At the early search stage, t is small and p_{tap} remains close to p_{max}, so taproot-based updates are more likely to be selected. This helps generate candidate solutions with relatively large search spans. As FEs increases, p_{tap} decreases gradually, and the fibrous-root update becomes more frequently activated. The search process then shifts toward local adjustment and refined exploitation. Through this adaptive probability schedule, IPRGO reduces the dependence on manually fixed switching parameters and makes the update mode more consistent with the optimization stage.

3.3 Stagnation-detection-driven local reset mechanism

In the encryption-parameter optimization process, the population may gradually concentrate around several similar candidate vectors. When the best fitness value remains unchanged for a number of consecutive evaluations, subsequent updates tend to generate limited improvement, and the search efficiency decreases. To alleviate this problem, IPRGO introduces a local reset mechanism triggered by stagnation detection. This mechanism only reinitializes a subset of inferior individuals, while the current high-quality individuals are preserved.

1) Stagnation criterion

Let the population at function evaluation t be denoted as $x_i(t), i = 1,\ldots,N$, where N is the population size and $f()$ represents the objective function.. Since IPRGO is formulated as a minimization algorithm, the best fitness value at evaluation t is defined as:

$$g(t) = \min_{1 \le i \le N} f\left(x_i(t)\right) \tag{7}$$

To monitor whether the search process continues to improve, a stagnation counter $stagCnt$ is introduced. It is updated according to:

$$stagCnt(t) = \begin{cases} 0, g(t) < g(t-1) \\ stagCnt(t-1)+1, g(t) \ge g(t-1) \end{cases} \tag{8}$$

When $g(t) < g(t-1)$, a better fitness value is obtained and the counter is reset to zero. Otherwise, the best fitness value does not improve, and the counter increases by one. If $stagCnt(t) \ge L_s$ the search is regarded as stagnant and the local reset operation is activated, where L_s denotes the predefined stagnation threshold.

2) Local reset

When $stagCnt(t) \ge L_s$, a partial reset strategy is performed to introduce new candidate positions. Instead of reinitializing the whole population, only a certain proportion of inferior individuals is selected for reset, while the better individuals are retained. Let r denote the reset ratio. The number of individuals selected for reset is calculated as $N_r = \lceil rN \rceil$, where $r \in (0,1]$, N is the population size, and $\lceil . \rceil$ denotes the ceiling function. Since the optimization objective is minimized in this study, all individuals are sorted in ascending order according to their fitness values. The sorted index sequence is expressed as $I = \{I_1, I_2, ..., I_N\}$, where I_1 corresponds to the best individual and I_N corresponds to the worst individual. Therefore, the reset index set is defined as $R = \{i_{N-N_r+1}, i_{N-N_r+2}, \ldots, i_N\}$, For each individual $j \in R$, its position is regenerated within the feasible search space as follows:

$$x_j = I_b + \xi_j \square \left(U_b - I_b\right) \tag{9}$$

where I_b and U_b are the lower and upper bounds of the search space, respectively. ξ_j is a D-dimensional random vector whose elements are uniformly distributed in $[0, 1]$, and $\square$ denotes element-wise multiplication.

Through this partial reset operation, new search directions can be introduced after stagnation occurs.

Meanwhile, the better individuals are retained, so the useful search information accumulated before stagnation is not discarded.

4 Proposed color image encryption scheme

4.1 Overall framework of the proposed encryption scheme

The proposed encryption scheme is organized around two connected stages: IPRGO-driven parameter optimization and optimization-guided image encryption. In the first stage, the parameters related to the four-dimensional Lorenz system and the perturbation layer are encoded into a unified control vector. IPRGO is then used to search for a parameter combination that can improve the encryption-oriented fitness value. In the second stage, the optimized control vector is introduced into the actual encryption process, rather than being used only as an external secret key. Specifically, the optimized chaos-layer parameters determine the generation of four-dimensional Lorenz sequences. These sequences are further transformed into row-column permutation indices and basic diffusion key streams. The perturbation-layer parameters then regulate key-stream reconstruction, including segment rearrangement, circular shifting, and channel fusion. After the final RGB key streams are obtained, the plain image is encrypted through synchronous row-column permutation across the three color channels and channel-coupled bidirectional diffusion. In this framework, parameter optimization, sequence generation, key-stream reconstruction, and permutation-diffusion execution are linked in a unified encryption procedure.

4.2 IPRGO-driven optimization of encryption control parameters

To link parameter optimization with the subsequent encryption operations, IPRGO is used to search the control parameter vector of the proposed encryption scheme. The control parameter vector is defined as $\theta = [\theta_c, \theta_k] \in R^{10}$,where $\theta_c = [\mathrm{x}_0, \mathrm{y}_0, \mathrm{z}_0, \mathrm{w}_0, \mathrm{r}]$ represents the chaos-layer parameters. These parameters are used to determine the initial states and control parameter of the four-dimensional Lorenz system. The perturbation-layer parameter subset is defined as $\theta_k = [\mathrm{seedRaw}, \mathrm{shiftRRaw}, \mathrm{shiftGRaw}, \mathrm{modeRaw}, \mathrm{KRaw}]$, where $\mathrm{seedRaw}$ controls the random seed used for segment-wise key-stream rearrangement, $\mathrm{shiftRRaw}$ and $\mathrm{shiftGRaw}$ determine the circular shift amounts of the R-channel and G-channel key streams, respectively, $\mathrm{modeRaw}$ specifies the fusion mode used to construct the B-channel key stream, and KRaw controls the segment scale in key-stream rearrangement. Therefore, θ_k determines how the basic key streams are further reorganized and reconstructed before diffusion.

Step 1: Population initialization

Let N be the population size, and let the position of individual i at iteration t be denoted as $\theta_i(t) = [\theta_{c,i}(t), \theta_{k,i}(t)]$ .During initialization, N candidate control vectors are randomly generated within the predefined parameter bounds:

$$\theta_i(0) = I_b + rand(1, D) \odot (U_b - I_b) \tag{10}$$

where D is the dimension of the control vector, I_b and U_b denote the lower and upper bound vectors of the parameter space, respectively. $rand(1, D)$ is a D-dimensional random vector uniformly distributed in [0,1], and $\odot$ denotes element-wise multiplication. In this way, the initial population contains different combinations of chaos-layer parameters and perturbation-layer parameters, providing the initial search candidates for IPRGO.

Step 2: Fitness evaluation

For each candidate control vector θ , the corresponding encryption process is executed first. Then, several security-related indicators are calculated from the generated cipher image. In this study, the fitness function is designed by combining information entropy, adjacent-pixel correlation, NPCR and UACI:

$$F(\theta) = \omega_1(8 - H(\theta)) + \omega_2\rho(\theta) + \omega_3\left|NPCR(\theta) - NPCR_0\right| + \omega_4\left|UACI(\theta) - UACI_0\right| \tag{11}$$

where, $H(\theta)$ denotes the information entropy of the cipher image generated under parameter vector θ, and $\rho(\theta)$ represents the adjacent-pixel correlation measure. $NPCR(\theta)$ and $UACI(\theta)$ are used to evaluate the sensitivity of the cipher image to slight plaintext changes. The coefficients $\omega_1, \omega_2, \omega_3, \omega_4$ are the weighting factors of different evaluation terms, while $NPCR_0$ and $UACI_0$ denote the ideal reference values of NPCR and UACI, respectively. Since IPRGO is implemented as a minimization algorithm, a smaller value of $F(\theta)$ indicates that the corresponding parameter vector produces a cipher image with entropy closer to 8, lower adjacent-pixel correlation, and differential metrics closer to their ideal values. Therefore, the fitness function provides a quantitative criterion for selecting encryption control parameters.

Step 3: Update the current best solution

After the fitness values of all candidate control vectors are obtained, the best solution in the current population is selected according to the minimization criterion. The best fitness value at iteration t is defined as:

$$g(t) = minF(\theta_i(t)), 1 \le i \le N \tag{12}$$

The corresponding best individual is expressed as:

$$\theta_{best}(t) = \arg(\min F(\theta_i(t))), 1 \le i \le N \tag{13}$$

To preserve the best solution found during the whole search process, the current best solution is further compared with the historical global best solution. The global best solution is updated as follows:

$$\theta_{gbest}(t) = \begin{cases} \theta_{best}(t), g(t) < F(\theta_{gbest}(t-1)) \\ \theta_{gbest}(t-1), otherwise \end{cases} \tag{14}$$

Through this update rule, IPRGO records the best parameter vector obtained so far and prevents high-quality solutions from being lost during subsequent population updates.

Step 4: Update candidate parameters

After the global best solution has been determined, the IPRGO update operator is applied to generate new candidate control vectors for the population. For individual i, the update process is expressed as:

$$\overline{\theta}_i(t+1) = u(\theta_i(t), \theta_{gbest}(t), t) \tag{15}$$

where, $u(\cdot)$ denotes the parameter update operator defined by the IPRGO search mechanism. Since the updated vector may fall outside the predefined feasible range, boundary correction is performed as follows:

$$\theta_i(t+1) = \prod_{[l,u]}(\overline{\theta}_i(t+1)) \tag{16}$$

where $\prod_{[l,u]}$ is the interval projection operator.

The corrected vector is then substituted into the encryption procedure to generate chaotic sequences, reconstruct key streams, and perform permutation-diffusion encryption. After that, its fitness value is recalculated. If the corrected vector obtains a smaller fitness value than the original vector, it replaces the original individual; otherwise, the original individual is retained. This selection rule is expressed as:

$$\theta_i(t+1) = \begin{cases} \theta_i(t+1), F(\theta_i(t+1)) < F(\theta_i(t)) \\ \theta_i(t), otherwise \end{cases} \tag{17}$$

Through this update-and-selection process, IPRGO keeps only parameter vectors that improve the encryption-oriented fitness value, while discarding ineffective candidate updates.

Step 5: Termination criterion and output of the optimal parameters

Steps 2–4 are repeated until the preset stopping condition is satisfied. In this study, the search process is terminated when the maximum number of function evaluations or the maximum number of iterations is reached. The final output of IPRGO is the historical global best control vector obtained during the whole optimization process, which is denoted as $\theta^* = \theta_{gbest}(T)$, where T denotes the final iteration or the final function-evaluation index. The optimized vector θ^* is then used as the encryption control parameter vector. Specifically, its chaos-layer component is used for four-dimensional Lorenz sequence generation, while its perturbation-layer component is used for key-stream rearrangement, circular shifting, and channel fusion in the subsequent encryption procedure.

4.3 Encryption procedure

After the optimal control vector is obtained by IPRGO, it is introduced into the concrete encryption process. The purpose of this subsection is to describe how the optimized parameters are used to drive chaotic sequence generation, key-stream reconstruction, RGB synchronous permutation, and channel-coupled bidirectional diffusion. The detailed encryption flow is shown in Fig.3.

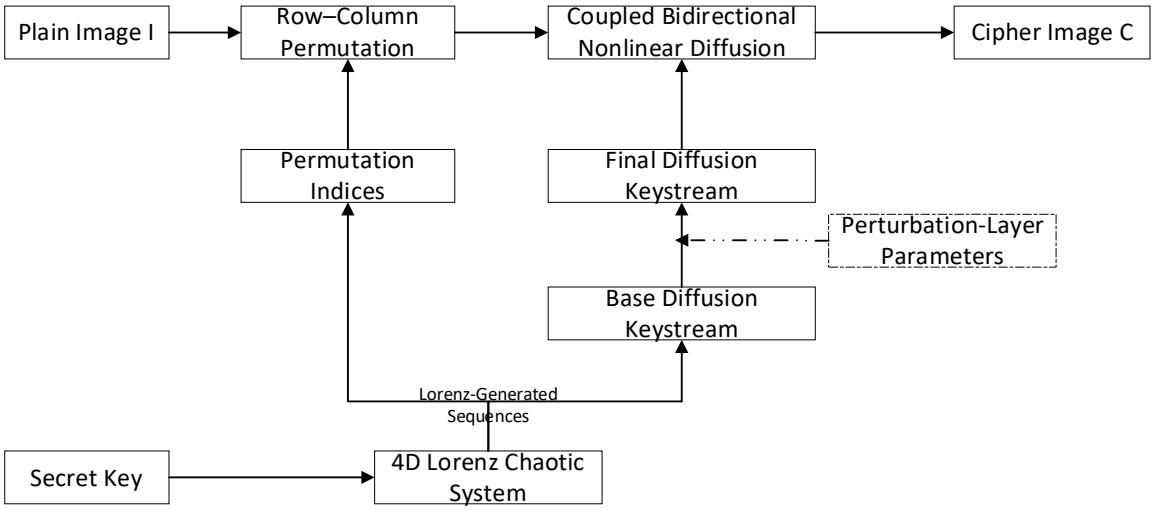


Fig.3 Flowchart of the specific encryption process

Step 1: Input the image and determine the parameters

Let the input color plain image be I, and its size be $M \times N \times 3$. The number of pixels in each color channel is $L=M \times N$. The optimized control parameter vector obtained from IPRGO is denoted as $\theta^* = [\theta_c^*, \theta_k^*]$, θ_c^* is used to initialize and control the four-dimensional Lorenz system, while θ_k^* is used in the subsequent key-stream modulation stage. In this way, the optimized result is directly connected with both chaotic sequence generation and key-stream reconstruction.

Step 2: Generate four-dimensional Lorenz chaotic sequences

The optimized chaos-layer parameters θ_c^* are assigned to the four-dimensional Lorenz system, and the system is numerically iterated to generate four state sequences $\{x_k\}, \{y_k\}, \{z_k\}, \{w_k\}$. To reduce the effect of the initial transient response, the first T values are removed from each sequence. The remaining values are then mapped into the interval $[0,1]$ as follows:

$$
\begin{aligned}
u_x(k) &= mod\left(\left|x_k\right|,1\right) \\
u_y(k) &= mod\left(\left|y_k\right|,1\right) \\
u_z(k) &= mod\left(\left|z_k\right|,1\right) \\
u_w(k) &= mod\left(\left|w_k\right|,1\right)
\end{aligned}
\tag{18}
$$

where, $u_x(k)$ and $u_y(k)$ are used to construct permutation indices, while $u_z(k)$ and $u_w(k)$ are used to construct the basic diffusion key streams.

Step 3: Construct permutation indices and basic diffusion key streams

After normalization, the chaotic sequences are converted into permutation indices and diffusion key streams. First, the first M elements of $u_x(k)$ and the first N elements of $u_y(k)$ are selected to generate the row and column ordering sequences. By sorting these elements in ascending order, the row permutation index and column permutation index are obtained as:

$$
\begin{aligned}
P_r &= argsort\left(u_x(1:M)\right) \\
P_c &= argsort\left(u_y(1:N)\right)
\end{aligned}
\tag{19}
$$

where P_r and P_c denote the row and column permutation indices, respectively. These two indices are later used to synchronously permute the R, G, and B channels.

The normalized sequences $u_z(k)$ and $u_w(k)$ are then transformed into byte-level key streams. The initial key streams for the R and G channels are generated as:

$$
\begin{aligned}
K_R^0(k) &= \left\lfloor 256 \cdot u_z(k) \right\rfloor \\
K_G^0(k) &= \left\lfloor 256 \cdot u_w(k) \right\rfloor \\
K_B^0(k) &= K_R(k) \oplus K_G(k)
\end{aligned}
\tag{20}
$$

After obtaining the basic key streams, the optimized perturbation-layer parameters θ_k^* are introduced to reconstruct the channel key streams before diffusion. The parameter KRaw specifies the segmentation scale, and the basic key streams are first divided into several subsequences according to this scale. Then, $\mathrm{seedRaw}$ is used to generate a pseudorandom rearrangement order, by which these subsequences are reordered. This operation changes the internal arrangement of the original key streams without altering their byte-level value range. Next, $\mathrm{shiftRRaw}$ and $\mathrm{shiftGRaw}$ are mapped to circular offsets for the R-channel and G-channel key streams, respectively. The shifted key streams provide different starting positions for the subsequent channel diffusion. Finally, $\mathrm{modeRaw}$ determines the generation rule of the B-channel key stream. According to its value, the B-channel key stream is produced by fusing the reconstructed R-channel and G-channel key streams through XOR operation or modular addition. After the above reconstruction operations, the final key streams K_R, K_G and K_B are obtained for the diffusion stage.

Step 4: Perform RGB synchronous row-column permutation

After the row and column permutation indices are obtained, the same pair of index mappings is applied to the three color channels. Different from independent channel permutation, this operation keeps the spatial rearrangement rule consistent across the R, G, and B channels. As a result, the pixel positions of the three channels are scrambled synchronously, which provides an aligned intermediate image for the subsequent channel-coupled diffusion stage. The permuted intermediate image is calculated as $I_p(i,j,c) = I\left(P_r(i), P_c(j), c\right)$. I_p represents the intermediate image after synchronous row-column permutation.

Step 5: Perform RGB coupled bidirectional nonlinear diffusion

After synchronous row-column permutation, the intermediate image is converted into one-dimensional channel sequences according to row-major scanning. The diffusion operation is then applied to the R, G, and B channels in sequence. For each channel, the permuted pixel sequence is denoted as $S=\{S(1),S(2),\ldots,S(L)\}$.

For the R channel, the forward diffusion process can be expressed as follows:

$$\begin{aligned} T_R(1) &= \Phi_1\left(\left(S_R(1)+K_R(1)+IV_R\right) mod\, 256\right) \\ T_R(k) &= \Phi_k\left(\left(S_R(k)+K_R(k)+T_R(k-1)\right) mod\, 256\right),\quad k=2,...,L \end{aligned} \tag{21}$$

where, IV_R is the initialization vector of the R channel, and $\Phi_k(\cdot)$ denotes a lightweight reversible affine nonlinear transformation.

After forward diffusion, reverse scanning is performed to produce the final R-channel cipher sequence:

$$\begin{aligned} C_R(L) &= \left(T_R(L)+K_R(L)+IV_R\right) mod\, 256 \\ C_R(k) &= \left(T_R(k)+K_R(k)+C_R(k+1)\right) mod\, 256, k=L-1,\ldots,1 \end{aligned} \tag{22}$$

Unlike independent diffusion of RGB channels, the proposed diffusion structure introduces channel-level coupling during the diffusion of the G and B channels. Specifically, the R-channel cipher sequence participates in the G-channel diffusion, and the G-channel cipher sequence further participates in the B-channel diffusion. For the G channel, the forward diffusion stage is expressed as:

$$\begin{aligned} T_G(1) &= \Phi_1\left(\left(S_G(1)+K_G(1)+IV_G+C_R(1)\right) mod\, 256\right) \\ T_G(k) &= \Phi_k\left(\left(S_G(k)+K_G(k)+T_G(k-1)+C_R(k)\right) mod\, 256\right), k=2,\ldots,L \end{aligned} \tag{23}$$

The diffusion form of the B channel is similar to that of the G channel, except that the coupling term is replaced with C_G .Through this progressively coupled bidirectional diffusion structure, a slight change in any pixel can propagate to subsequent pixels and further diffuse to other channels, thereby significantly enhancing the avalanche effect and resistance to differential attacks of the system.

Step 6: Output the cipher image

After the coupled bidirectional diffusion is completed, the three diffused channel sequences are converted back to two-dimensional matrices with size M × N. The reconstructed R, G, and B components are then combined along the channel dimension to form the final cipher image C. At this point, the spatial permutation and channel-coupled pixel-value diffusion have both been completed, and the output image no longer retains the visible structure of the original plain image.

4.4 Decryption procedure

The decryption procedure is carried out by applying the inverse operations in the reverse order of encryption. First, the cipher image is separated into R, G, and B channel sequences. Then, the inverse diffusion operation is performed sequentially from the B channel to the G channel and then to the R channel, which is opposite to the channel-coupling order used in encryption. After the permuted intermediate image is recovered, inverse column permutation and inverse row permutation are applied according to the inverse mappings of the column and row permutation indices. Finally, the restored R, G, and B channels are merged to obtain the recovered plain image.

For the R channel, the inverse diffusion first recovers the intermediate sequence T_R from the cipher sequence C_R . The reverse-scanning stage is expressed as:

$$T_R(L) = \left(C_R(L) - K_R(L) - IV_R\right) mod\, 256$$
$$T_R(k) = \left(C_R(k) - K_R(k) - C_R(k+1)\right) mod\, 256, k = L-1, \ldots, 1 \quad (24)$$

Then, the permuted R-channel sequence S_R is recovered by applying the inverse nonlinear transformation:

$$S_R(1) = \left(\Phi_k^{-1}\left(T_R(1)\right) - K_R(1) - IV_R\right) mod\, 256$$
$$S_R(k) = \left(\Phi_k^{-1}\left(T_R(k)\right) - K_R(k) - T_R(k-1)\right) mod\, 256, k = 2, \ldots L \quad (25)$$

The inverse diffusion of the G and B channels follows the same rule as that of the R channel, except that the corresponding channel-coupling terms are removed according to the reverse coupling order. Since the row-column permutation is reversible and the affine nonlinear transformation is invertible under modulo-256 arithmetic, the whole encryption process can be reversed without information loss. Therefore, the recovered image satisfies $D\left(E(I)\right) = I$.

5 Experimental results and security analysis

5.1 Experimental settings

The experimental evaluation was conducted on three standard color images, namely Lena, Peppers, and House. These images were used to verify the encryption effect, statistical security, key sensitivity, differential sensitivity, and robustness of the proposed scheme.

In the chaotic sequence generation stage, the fixed parameters of the four-dimensional Lorenz system were set as $a = 10$, $b = 8/3$ and $c = 28$. The numerical step size was $h = 0.005$, and the first 3000 iterative values were discarded to reduce the influence of transient states. The initialization vectors used in the diffusion stage were set as $IV_R = 11$, $IV_G = 77$ and $IV_B = 173$.

The optimized control vector Θ is 10-dimensional. The search ranges of the chaos-layer parameters were set as $x_0, y_0, z_0, w_0 \in \left(10^{-6}, 1-10^{-6}\right)$ and $r \in [-1.52, -0.06]$, while all perturbation-layer parameters were limited to $[0,1)$. During the optimization stage, the population size of IPRGO was set to $N = 15$, the maximum number of fitness evaluations was $MaxFEs = 160$, and each experiment was independently repeated $R = 30$. The adaptive probability bounds were set as $p_{max} = 0.75$ and $p_{min} = 0.25$, The stagnation threshold and restart ratio were set as $stagLimit = 18$ and $restartFrac = 0.25$, respectively. In the fitness function, the weights were assigned as $w_1 = 1.0$, $w_2 = 0.6$, $w_3 = 1.6$ and $w_4 = 0.8$. These settings were kept fixed in all following experiments.

5.2 Performance verification of IPRGO on CEC2017 benchmark functions

5.2.1 Benchmark settings and evaluation metrics

Before applying IPRGO to encryption parameter optimization, its numerical search performance was examined on the CEC2017 benchmark function suite. The test set contains 30 benchmark functions, denoted as F1–F30, and two dimensions, $D = 10$ and $D = 30$, were considered in the experiments. PSO, GWO, COA, PRGO, and the proposed IPRGO were selected for comparison. To keep the experimental configuration consistent, all algorithms were tested under the same population size and function-evaluation budget. The population size was set to 30, and the maximum number of function evaluations was defined as $MaxFEs = 10000 \times D$, For each benchmark function, every algorithm was independently executed 10 times using different random seeds. The best objective value obtained at the end of each run was recorded for statistical analysis.

The mean value and standard deviation were calculated from the 10 independent runs. The mean value

was used to describe the average optimization accuracy of an algorithm on a given function, while the standard deviation was used to reflect the fluctuation of the obtained results. In addition, three summary indicators were further reported: best-performing cases, average rank, and minimum-std cases. The best-performing cases count how many functions an algorithm obtains the lowest mean value on; the average rank reflects its overall position among all comparison algorithms; and the minimum-std cases indicate how often the algorithm achieves the smallest result fluctuation. These indicators jointly describe the optimization accuracy, comprehensive ranking, and running stability of each algorithm.

5.2.2 Statistical analysis of benchmark results

The statistical results on the CEC2017 benchmark functions are presented in Table 1. For each function, the mean value reflects the optimization accuracy obtained after repeated runs, while the standard deviation indicates the stability of the search results. A lower mean value means that the algorithm can reach a better objective value on the corresponding function, and a smaller standard deviation means that the algorithm is less affected by random initialization.

From the results under $D = 10$, IPRGO obtains competitive mean values on most benchmark functions and achieves a favorable average ranking among the compared algorithms. This indicates that the adaptive growth-mode probability and the stagnation-detection reset mechanism can improve the search behavior of PRGO under low-dimensional conditions. In particular, when the original PRGO shows result fluctuation or insufficient search refinement, IPRGO can maintain a more stable optimization process by adjusting the update mode and reintroducing candidate diversity after stagnation.

When the dimension increases to $D = 30$, the search difficulty becomes higher because the feasible space expands significantly. Under this condition, the advantage of IPRGO is still observable in the overall ranking and stability indicators. Although no single algorithm can obtain the best result on all benchmark functions, IPRGO shows a balanced performance in terms of mean value, average rank, and standard deviation. This suggests that the proposed improvement does not merely enhance a single search behavior, but helps PRGO coordinate global exploration, local refinement, and stagnation recovery more effectively.

Overall, the benchmark results verify that IPRGO has better comprehensive search capability than the original PRGO. The improvement is reflected not only in the optimization accuracy, but also in the stability of repeated runs. Therefore, IPRGO is suitable for the subsequent encryption parameter optimization task, where the control vector contains multiple coupled variables and each fitness evaluation requires executing the encryption process.

Table 1 Simplified statistical results of IPRGO and comparison algorithms on the CEC2017 test functions

Dimension	Statistical metric	PSO	GWO	COA	PRGO	IPRGO
D=10	Best-performing cases	3	3	13	0	12
	Average rank	3.25	2.83	2.18	4.93	1.8
	Minimum-std cases	2	0	13	1	14
D=30	Best-performing cases	2	11	2	0	15
	Average rank	3.20	2.20	2.90	5.00	1.70
	Minimum-std cases	1	3	12	1	13

5.2.3 Convergence performance analysis

To further examine the search trajectories of different algorithms, six representative functions, including F1, F6, F12, F17, F22, and F24, are selected from the CEC2017 benchmark suite. The average best-so-far convergence curves under $D = 30$ are obtained from 10 independent runs, as shown in Fig.4.

From the overall trend, IPRGO presents faster descent and better search behavior on most selected functions. For F1, F12, and F22, the objective values obtained by IPRGO decrease rapidly at the early stage and remain lower than those of the comparison algorithms during most subsequent iterations. The final values on

these functions are also clearly better than those of the other algorithms. This result shows that the two improvement strategies introduced into PRGO can strengthen its ability to locate promising search regions and maintain effective optimization along the iteration process.

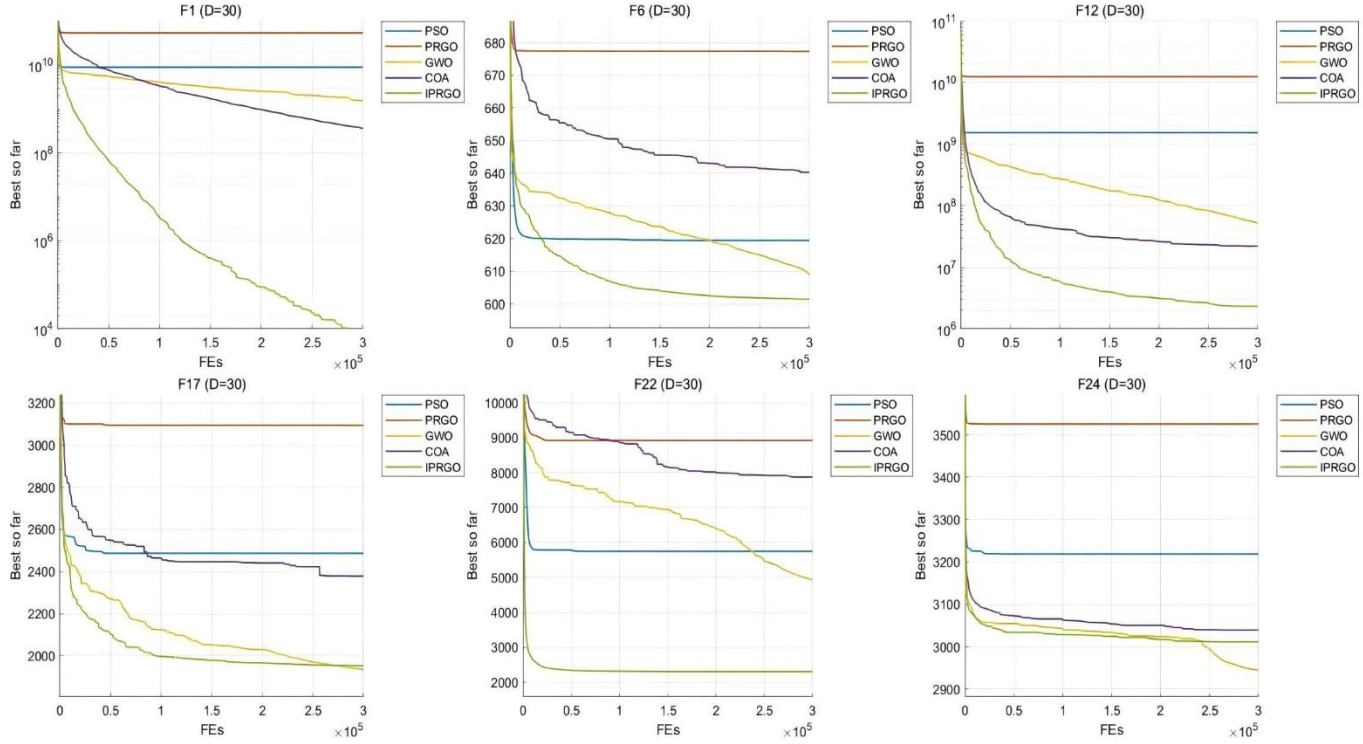


Fig.4 Average convergence curves of different algorithms on six representative test functions

For F6, the final objective values of different algorithms are relatively close. Even in this case, IPRGO reaches a lower objective level within fewer function evaluations and keeps a stable convergence tendency in the later stage. Its final result is better than those of PSO, PRGO, and COA, and is slightly better than that of GWO, indicating that IPRGO still maintains good search efficiency and convergence stability on this function.

For F17, IPRGO and GWO obtain close final results. However, IPRGO decreases more quickly at the beginning and keeps lower function values during most iterations. This means that IPRGO still shows an advantage in the convergence process, although the final leading margin is not as obvious as that on F1, F12, and F22. This also suggests that F17 requires stronger late-stage refined search capability.

For F24, IPRGO also enters a lower-value region quickly in the early search stage. However, its final function value is slightly higher than that of GWO, and it does not obtain the best final result on this function. This result indicates that IPRGO has good early-stage convergence speed, but its local exploitation ability still needs further improvement on a few individual functions.

In summary, the proposed algorithm does not absolutely outperform all comparison algorithms on every representative function. Nevertheless, its convergence behavior on most selected functions is better than that of the original PRGO and remains competitive against PSO, GWO, and COA. The convergence-curve results further support the effectiveness of IPRGO from the perspective of dynamic search behavior and provide a basis for using it in the subsequent image-encryption parameter optimization task.

5.3 Visual results of encryption and decryption

To evaluate the actual encryption effect of the proposed scheme, IPRGO is first used to optimize the 10-dimensional control parameter vector. The results of 30 independent runs show that the obtained optimal fitness values fluctuate within a small range, which indicates that IPRGO can provide stable search results under the preset function-evaluation budget. The final optimized parameter vector is obtained as

$\theta^* = [0.589768, 0.191875, 0.176899, 0.325162, -0.552684, 0.566744, 0.0240423, 0.248052, 0.0383549, 0.698351]$.

The average convergence curve of IPRGO is shown in Fig.5. The fitness value decreases quickly in the early stage, showing that the population can approach a better parameter region within a limited number of evaluations. As the search proceeds, the curve gradually becomes stable, indicating that the algorithm enters a relatively steady refinement stage around the current promising solution. This trend is consistent with the design of the adaptive growth-mode probability adjustment and the stagnation-detection-driven local reset mechanism.

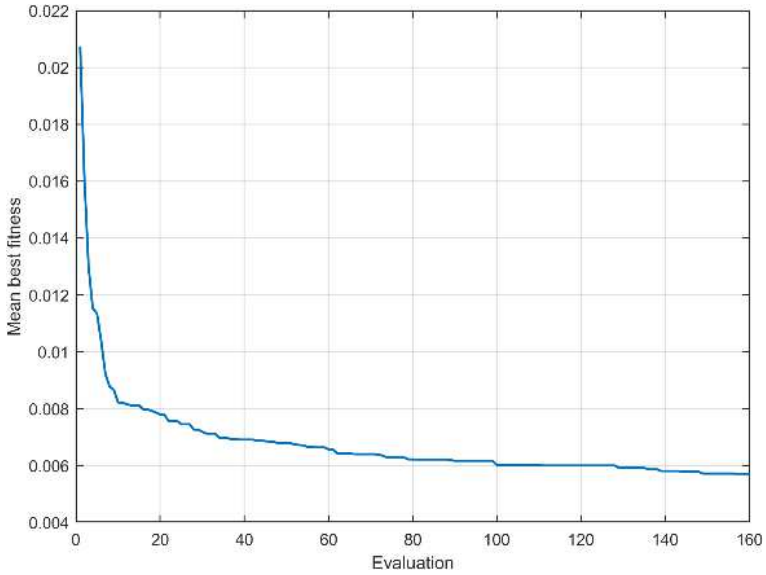


Fig.5 Average convergence curve of the IPRGO algorithm

Using the optimized parameter vector, encryption and decryption experiments are carried out on Lena, Peppers, and House. The visual results are presented in Fig.6.

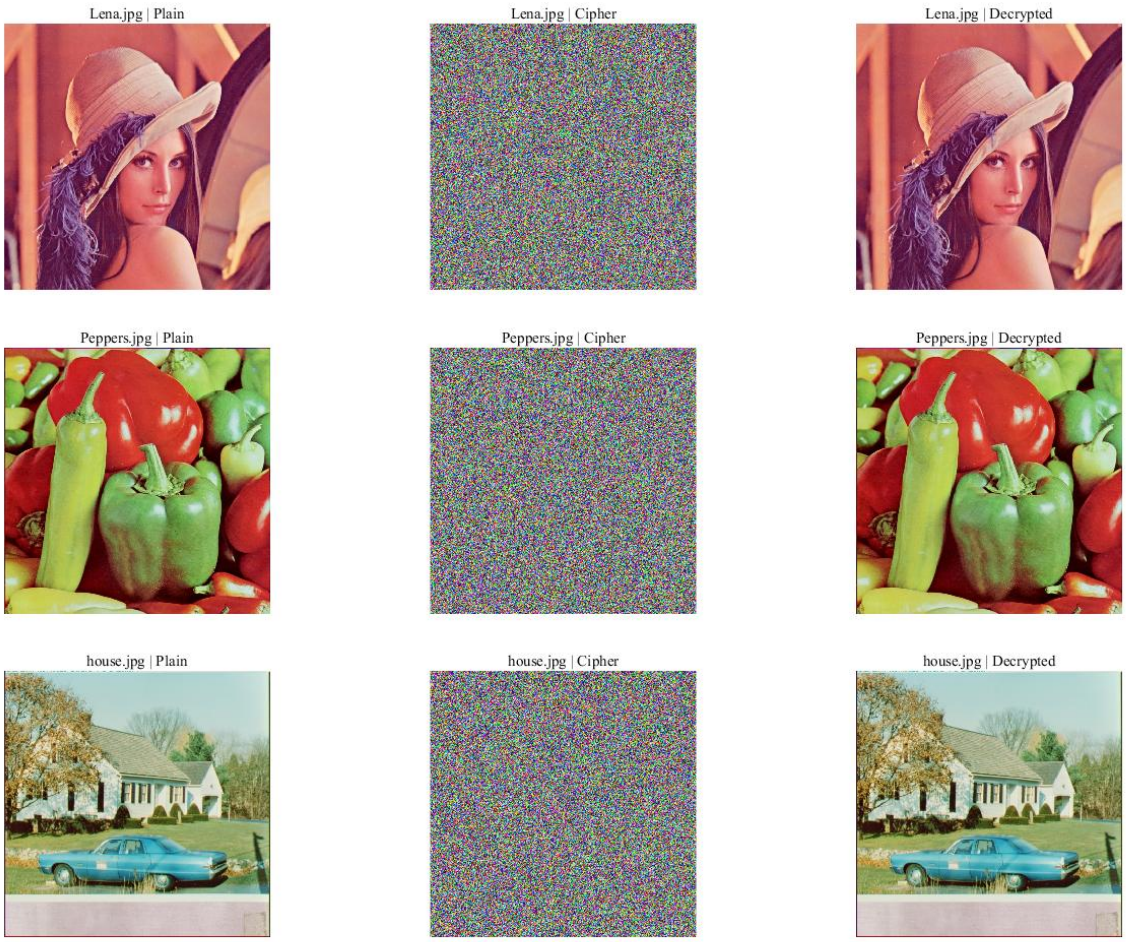


Fig.6 Encryption and decryption results of color images

After applying the proposed encryption scheme, the ciphertext images lose the identifiable structural characteristics of the original contents, making visual interpretation infeasible. This observation indicates that the designed permutation and channel-coupled diffusion mechanisms successfully disrupt the visual patterns contained in the plaintext images. When the correct key is used for decryption, the recovered images are visually consistent with the corresponding plain images, which verifies the reversibility of the encryption process and the correctness of the decryption procedure.

Overall, the visual results show that the proposed method can obtain a stable optimized parameter vector through IPRGO and can achieve effective visual concealment and correct decryption recovery on typical color images. These results provide a direct visual basis for the subsequent statistical and security analyses.

5.4 Security analysis

5.4.1 Histogram analysis

Histogram analysis is performed to examine the variation of pixel intensity distributions before and after encryption. Nonuniform concentration of ciphertext intensities may reveal residual statistical characteristics of the original image, which could provide potential information leakage[25]. Therefore, a desirable ciphertext should exhibit a balanced intensity distribution among different channels to minimize statistical distinguishability.

Fig.7 compares the histograms of the R, G, and B channels of Lena before and after encryption. In the plain image, the pixel values are not evenly distributed. Several intervals contain dense pixel occurrences, and the three channels show different distribution shapes, which is consistent with the visual structure and color

composition of the original image. After encryption, this distribution pattern is largely removed. The histograms of the three cipher-image channels become close to a flat form, and the pixel values spread over the full gray range from 0 to 255 without obvious local aggregation.

This result indicates that the proposed encryption process effectively reshapes the gray-level statistics of the original image. The synchronous permutation changes the spatial arrangement of pixels, while the channel-coupled diffusion further redistributes pixel values across the RGB channels. As a result, the cipher image no longer preserves the histogram characteristics of the plaintext image, which weakens the possibility of extracting useful statistical information from histogram observation.

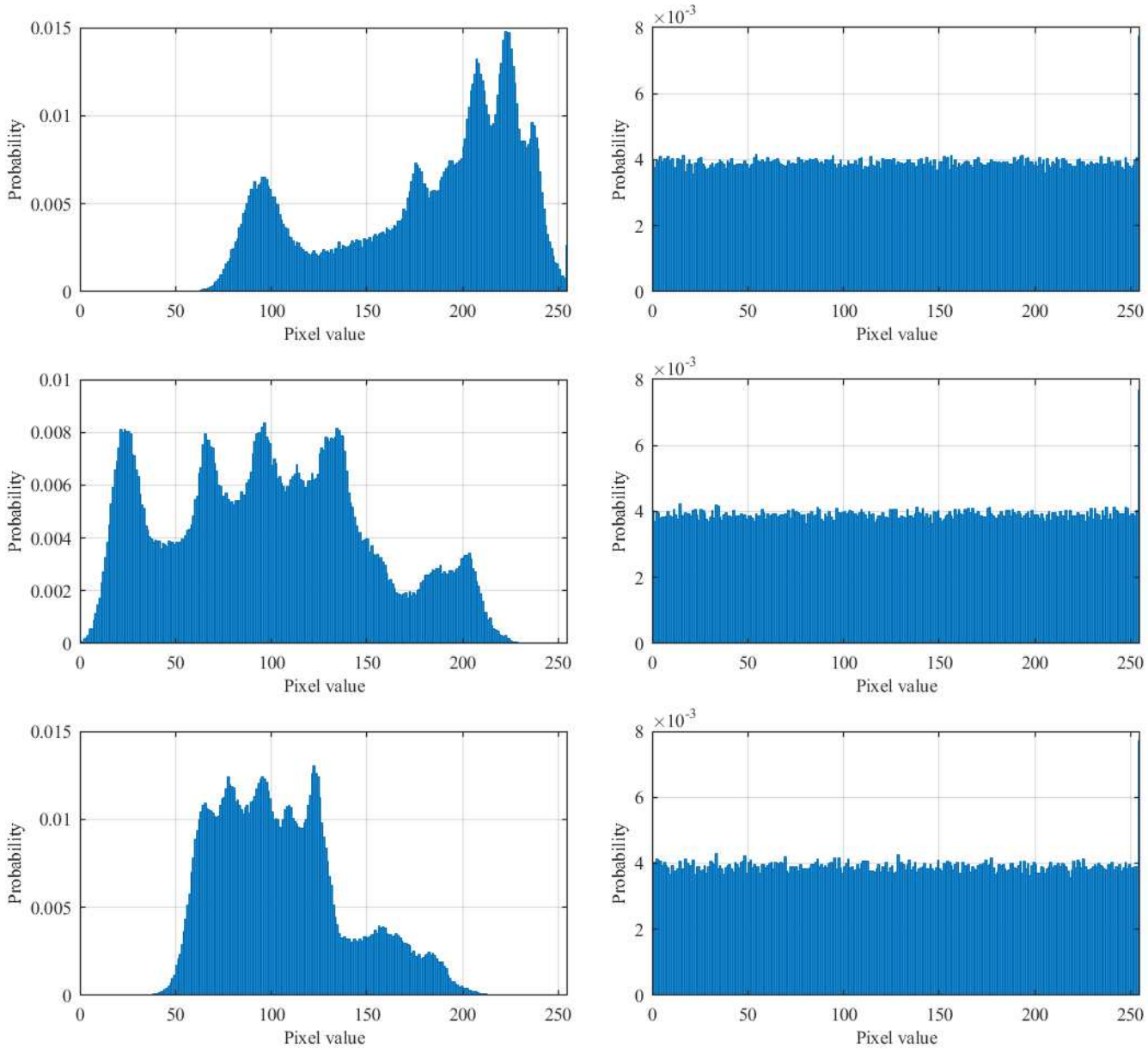


Fig.7 Histogram comparison of the R, G, and B channels of the Lena color image before and after encryption

5.4.2 Information entropy analysis

Information entropy is adopted to further examine the randomness of the encrypted images. In an 8-bit image, the entropy value approaches its upper bound when pixel values appear with nearly equal probability[26]. Therefore, after encryption, the entropy of each cipher-image channel should be as close as possible to 8. The calculation formula is given as:

$$En(p) = -\sum_{p=0}^{255} p_p \log_2 p_p \tag{26}$$

where, p_p denotes the probability of occurrence of a pixel value.

Table 2 reports the entropy values of the R, G, and B channels for Lena, Peppers, and House. Before encryption, the entropy values remain below the ideal level because natural images contain nonuniform structures and redundant information. Following encryption, the entropy values of all RGB channels approach the theoretical maximum, indicating a highly uncertain ciphertext distribution. Specifically, the cipher-image entropy values of Lena reach 7.9994, 7.9993, and 7.9994 for the R, G, and B channels, respectively. Similar results are also obtained for Peppers and House.

These values indicate that the proposed encryption method makes the pixel distribution of each channel highly random. Combined with the histogram results in Fig.7, the entropy results further confirm that the original statistical distribution is effectively suppressed after encryption. Consequently, the encrypted images provide limited statistical clues that can be exploited for cryptanalysis.

Table 2 Information entropy of the RGB channels of Lena, Peppers, and House images before and after encryption

Test image	Plain image			Cipher image		
	R channel	G channel	B channel	R channel	G channel	B channel
Lena	7.2634	7.5899	6.9854	7.9994	7.9993	7.9994
Peppers	7.3538	7.5840	7.1286	7.9992	7.9993	7.9993
House	7.4541	7.2685	7.4927	7.9992	7.9993	7.9993

5.4.3 Adjacent pixel correlation analysis

The spatial dependency between neighboring pixels is analyzed to determine whether encryption effectively breaks the intrinsic relationships of the plaintext. Due to texture continuity and local structures, neighboring pixels in natural images usually exhibit strong similarity. If this relationship is still present in the cipher image, part of the spatial structure may be inferred from statistical observation[27,28]. Therefore, the correlation of adjacent pixels is calculated in the horizontal, vertical, and diagonal directions. The correlation coefficient is defined as:

$$\begin{aligned} Corr_{p_1,p_2} &= \frac{\mathrm{cov}(p_1,p_2)}{\sigma_{p_1}\sigma_{p2}} \\ \mathrm{cov}(p_1,p_2) &= E((p_1-\mu_{p_1})(p_2-\mu_{p2})) \end{aligned} \tag{27}$$

where, p_1 and p_2 denote two adjacent pixels, μ_{p_1} and μ_{p_2} are their mean values, σ_{p_1} and σ_{p_2} are their standard deviations, and $\mathrm{E}()$ denotes the mathematical expectation.

Table 3 gives the correlation coefficients of the R, G, and B channels of Lena before and after encryption. In the plain image, the coefficients in the three directions are all close to 1, indicating that adjacent pixels have strong spatial consistency. After encryption, the coefficients are reduced to values around 0. The largest absolute value is only 0.00154, showing that the spatial dependency among neighboring pixels has been largely removed.

Table 3 Adjacent-pixel correlation coefficients of the RGB channels of the Lena image in different directions before and after encryption

Direction	Plain image			Cipher image		
	R channel	G channel	B channel	R channel	G channel	B channel
Horizontal	0.97534	0.97482	0.95320	0.00031	-0.000083	-0.00028
Vertical	0.98709	0.98724	0.97413	-0.00029	-0.00039	-0.00037
Diagonal	0.96343	0.96298	0.93340	-0.00154	-0.00043	-0.000072

The scatter plots Fig.8 provide a more intuitive observation. The plain-image points are concentrated along a diagonal band, which means that adjacent pixels tend to change together. For the cipher image, the points are dispersed over the plane without a clear linear trend. This distribution agrees with the numerical results in Table 3 and confirms that the proposed synchronous permutation and channel-coupled diffusion effectively break the neighborhood relationship of the original image.

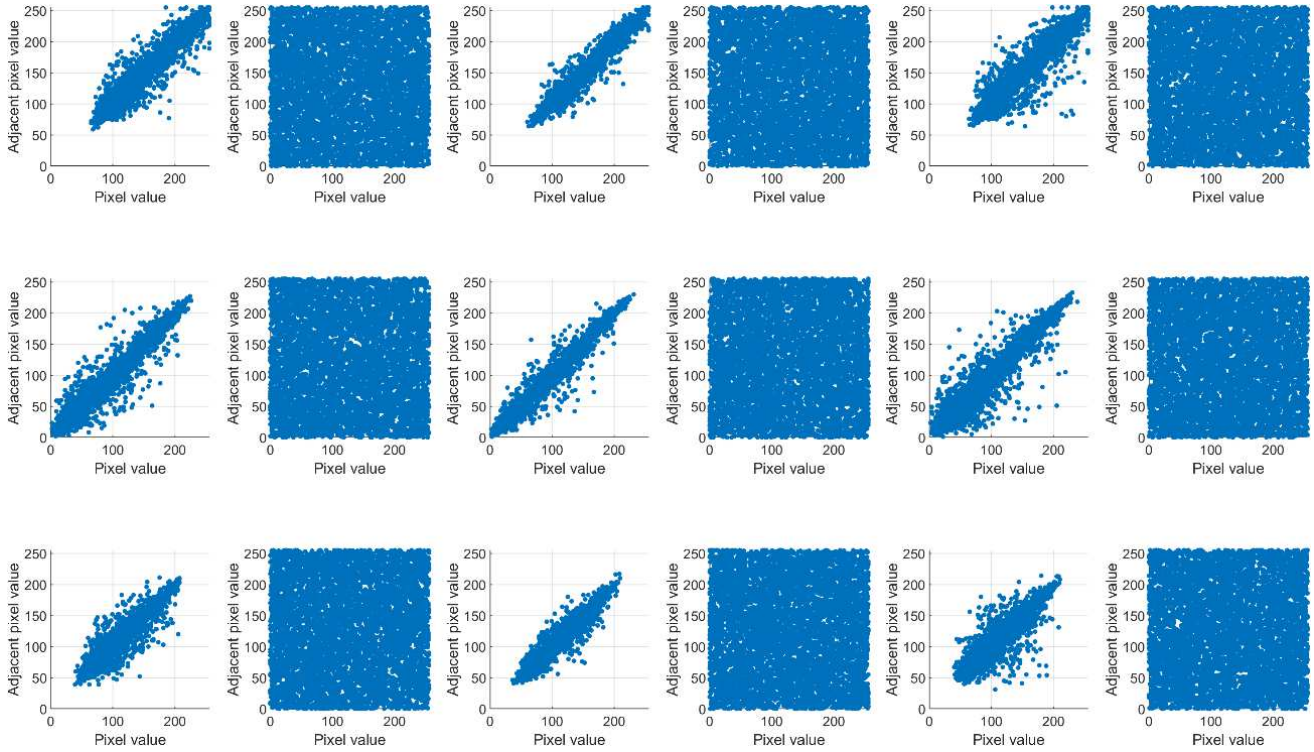


Fig.8 Comparison of adjacent-pixel correlation scatter plots of the R, G, and B channels of the Lena color image in the horizontal, vertical, and diagonal directions for the plain image and cipher image

5.4.4 Differential attack analysis

A robust image encryption scheme should generate significant ciphertext variations when the input image undergoes minor modifications. In this experiment, a single pixel of the plain image is modified, and the original and perturbed images are encrypted separately under the same key. The difference between the two generated cipher images is then evaluated by the number of pixels change rate (NPCR) and unified average changing intensity (UACI). NPCR quantifies the proportion of changed ciphertext pixels, whereas UACI measures the average intensity difference introduced by plaintext perturbations.[29]. The two indicators are calculated as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%$$
$$D(i,j) = \begin{cases} 0, p(i,j) = p'(i,j) \\ 1, otherwise \end{cases} \tag{28}$$

where, $p(i,j)$ and $p'(i,j)$ denote the pixels of two different cipher images, respectively, and M and N are the image dimensions.

UACI measures the average intensity difference between the two cipher images. It is calculated as:

$$UACI = \frac{\sum_{i,j} | p(i,j) - p'(i,j) |}{255 \times M \times N} \times 100\% \tag{29}$$

For 8-bit images, the theoretical reference values of NPCR and UACI are approximately 99.609% and 33.464%, respectively. Table 4 lists the results obtained from Lena, Peppers, and House. For Lena, the three-channel NPCR values are 99.609%, 99.621%, and 99.597%, while the corresponding UACI values are 33.466%, 33.473%, and 33.447%. The results for Peppers and House also remain near the reference range.

These results indicate that the plaintext perturbation is not limited to a local region after encryption. Instead, it is propagated to most positions of the cipher image through RGB synchronous permutation and channel-coupled bidirectional diffusion. Therefore, the proposed method shows strong plaintext sensitivity and can reduce the effectiveness of differential analysis.

Table 4 NPCR and UACI of the RGB channels of three test images under single-pixel perturbation

Test image	NPCR			UACI		
	R channel	G channel	B channel	R channel	G channel	B channel
Lena	99.609%	99.621%	99.597%	33.466%	33.473%	33.447%
Peppers	99.534%	99.607%	99.586%	33.461%	33.465%	33.451%
House	99.593%	99.615%	99.583%	33.497%	33.452%	33.484%

5.4.5 Key-space analysis

The size of the key space is evaluated to determine the feasibility of resisting exhaustive key searching attacks. In image encryption, a key space larger than 2^{100} is generally regarded as sufficient for resisting brute-force attacks[30].

In the proposed scheme, the effective key is determined by the optimized 10-dimensional control vector. This vector contains the initial states and control parameter of the four-dimensional Lorenz system, together with the perturbation-layer parameters used for key-stream rearrangement, circular shifting, and channel fusion. These parameters are not independent of the encryption procedure. Instead, they affect chaotic sequence generation, permutation-index construction, key-stream reconstruction, and diffusion execution. Therefore, all components of the control vector should be included in the key-space estimation.

If the numerical precision of each continuous parameter is 10^{-14}, the total key space can be estimated as $10^{140.16} \approx 2^{465.62}$.The obtained key space is far beyond the empirical security threshold of 2^{100}. Thus, the probability of finding the correct control vector by exhaustive search is extremely low, indicating that the proposed method has sufficient resistance to brute-force attacks from the perspective of key-space size.

5.4.6 Key sensitivity analysis

Key sensitivity is examined by decrypting the cipher images with slightly perturbed keys. If the encryption scheme is sufficiently sensitive, a small deviation in any effective key component should prevent the recovery of meaningful plaintext information.[31]. In this experiment, the difference between the original image and the image decrypted with an incorrect key is evaluated by mean square error (MSE), peak signal-to-noise ratio (PSNR), and signal-to-noise ratio (SNR), which are defined as:

$$\begin{aligned} MSE &= \frac{1}{MNC}\sum (I - I_w)^2 \\ PSNR &= 10\log_{10}\frac{255^2}{MSE} \\ SNR &= 10\log_{10}\frac{\sum I^2}{\sum (I - I_w)^2} \end{aligned} \tag{30}$$

where I denotes the original plaintext image, and I_w denotes the image decrypted using a perturbed key. M, N and C are the height, width, and number of channels of the image, respectively. A large MSE together with

low PSNR and SNR values means that the wrong-key decrypted image is far from the original image and cannot provide useful plaintext information.

Two representative key components are selected for perturbation: the chaos-layer parameter x0 and the perturbation-layer parameter *seedRaw*. Specifically, x_0 is changed by 10^{-14} and *seedRaw* is changed by 10^{-16}, while all other key parameters remain fixed. Fig.9 shows the corresponding decryption results, and Table 5 reports the quantitative values.

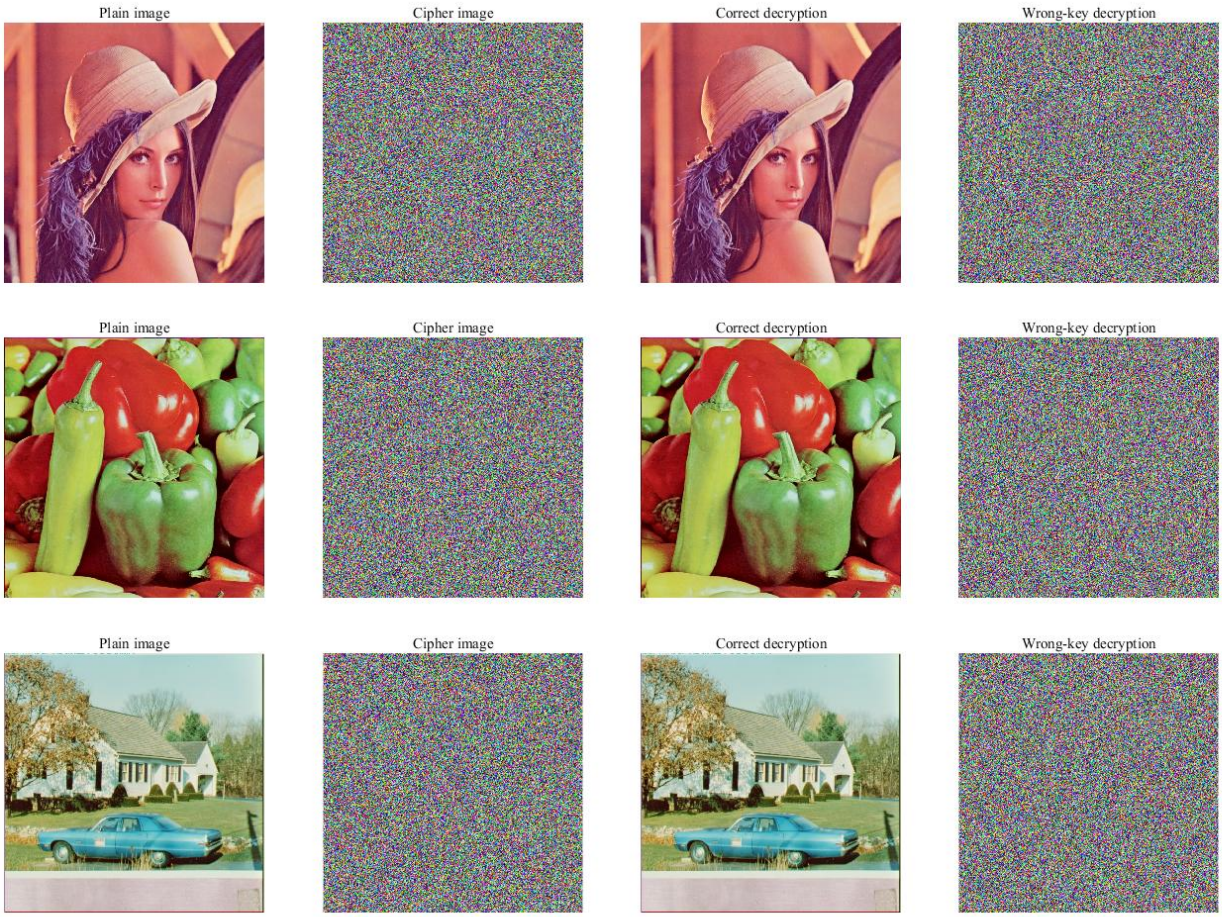


Fig.9 Comparison of correct-key and wrong-key decryption results

Table 5 Wrong-decryption evaluation results under different key perturbation conditions

Test image	Perturbed parameter	MSE	PSNR	SNR
Lena		8917.3	8.628dB	3.489dB
Peppers	x_0	10072.0	8.100dB	2.168dB
House		9209.4	8.488dB	4.687dB
Lena		8941.6	8.617dB	3.478dB
Peppers	*SeedRaw*	10087.0	8.093dB	2.162dB
House		9209.3	8.489dB	4.687dB

As shown in Fig.9, decryption with the correct key restores the original visual content, whereas the results obtained with the perturbed keys appear as random-like images. No recognizable edge, texture, or object contour can be observed. The numerical results in Table 5 are consistent with this visual observation. For both x_0 and *seedRaw* perturbations, the MSE values remain at a high level, while PSNR and SNR stay very low. This means that even a tiny mismatch in either the chaotic-system parameter or the key-stream reconstruction

parameter causes decryption failure. Therefore, the proposed method is highly sensitive to key perturbation and does not allow approximate-key recovery.

5.4.7 Robustness analysis under salt-and-pepper noise and cropping attacks

In practical transmission or storage, cipher images may be affected by random noise or partial data loss. To examine the recovery ability of the proposed scheme under such degradation, two types of attacks are considered: salt-and-pepper noise and block cropping. After the cipher image is attacked, the damaged cipher image is decrypted with the correct key.[32]. The recovered image is then compared with the original plain image using PSNR and SNR. The definitions of MSE, PSNR, and SNR are the same as those in the key-sensitivity experiment, except that the comparison object is replaced by the image recovered from the attacked cipher image.

In the salt-and-pepper noise experiment, the noise densities are set to 0.005, 0.01, and 0.02. In the cropping experiment, the block loss ratios are set to 0.01, 0.02, and 0.04. The quantitative results are listed in Tables 6 and 7. As the attack intensity increases, both PSNR and SNR show a decreasing trend, which means that stronger cipher-image degradation causes more severe distortion in the decrypted result. Under the tested settings, the block-cropping cases generally retain higher PSNR and SNR values than the salt-and-pepper noise cases. This is because block cropping destroys a continuous local region, whereas salt-and-pepper noise introduces scattered pixel-level interference over the entire cipher image.

Table 6 Robustness results under different salt-and-pepper noise densities

Test image	d=0.005		d=0.01		d=0.02	
	SNR	PSNR	SNR	PSNR	SNR	PSNR
Lena	15.044dB	20.184dB	12.161dB	17.300dB	9.415dB	14.554dB
Peppers	13.368dB	19.300dB	10.451dB	16.382dB	7.758dB	13.690dB
House	15.972dB	19.773dB	13.129dB	16.930dB	10.382dB	14.183dB

Table 7 Robustness results under different block cropping ratios

Test image	p=0.01		p=0.02		p=0.04	
	SNR	PSNR	SNR	PSNR	SNR	PSNR
Lena	19.453dB	24.592dB	16.252dB	21.392dB	13.680dB	18.820dB
Peppers	18.245dB	24.176dB	15.055dB	20.987dB	12.343dB	18.275dB
House	20.629dB	24.430dB	17.522dB	21.324dB	14.953dB	18.754dB

Fig.10 shows the recovered Lena images under different attack strengths. Although image quality decreases as the noise density or cropping ratio increases, the main facial contour and basic visual content can still be observed in the decrypted results. The degradation is more obvious under salt-and-pepper noise, where random pixel disturbances are distributed globally. These results indicate that the proposed method has a certain tolerance to cipher-image damage, but the final recovery quality is still related to the attack type and degradation level.



Fig.10 Decryption results of the Lena cipher-image under different salt-and-pepper noise densities and block cropping ratios

5.5 Comparative analysis with existing algorithms

To further investigate the competitiveness of the proposed scheme, comparisons with representative image encryption approaches are conducted based on four security indicators: entropy, neighboring-pixel correlation, NPCR, and UACI. The corresponding results are summarized in Tables 8–11

As reported in Table 8, the proposed method achieves an average entropy of 7.9994, indicating that the generated ciphertext maintains a highly balanced intensity distribution. Compared with the reference algorithms, the obtained value approaches the theoretical upper limit of an 8-bit image channel, demonstrating enhanced uncertainty in ciphertext pixels.

The comparison results of adjacent-pixel correlation are presented in Table 9 The proposed scheme achieves a mean absolute correlation coefficient of 0.00042 across three directions, outperforming the listed comparison methods. This observation verifies that the designed permutation and channel-coupled diffusion mechanisms effectively disrupt the spatial relationships inherited from the original image.

According to Table 10 and 11, the proposed scheme obtains average NPCR and UACI values of 99.609% and 33.462%, respectively. These results indicate that slight modifications in plaintext can induce considerable variations in ciphertext, providing strong resistance against differential analysis.

Overall, the proposed scheme achieves a balanced security performance by jointly maintaining high ciphertext uncertainty, weak spatial dependency, and strong sensitivity to plaintext variations. Rather than relying on improvement in a single indicator, the comparative results demonstrate its comprehensive capability across multiple security evaluations.

Table 8 Comparison of the proposed method and reference methods in terms of RGB-channel and average information entropy

Method	R channel	G channel	B channel	Average
Proposed method	7.9994	7.9993	7.9994	7.9994

Ref.[33]	7.9948	7.9950	7.9955	7.9951
Ref.[34]	7.9985	7.9986	7.9986	7.9982
Ref.[35]	7.9975	7.9974	7.9972	7.9974

Table 9 Comparison of the average absolute values of correlation coefficients in the horizontal, vertical, and diagonal directions between the proposed method and reference methods

Method	Horizontal	Vertical	Diagonal	Average
Proposed method	0.00022	0.00035	0.00068	0.00042
Ref.[33]	0.0035	0.0054	0.0004	0.0031
Ref.[36]	0.0032	0.0023	0.0014	0.0023
Ref.[37]	0.0032	0.0021	0.00082	0.0020

Table 10 Comparison of the proposed method and reference methods in terms of NPCR of the RGB channels and the average value

Method	R channel	G channel	B channel	Average
Proposed method	99.609%	99.621%	99.597%	99.609%
Ref.[3]	99.6099%	99.6129%	99.6102%	99.6110%
Ref.[35]	99.5605%	99.5926%	99.6231%	99.5921%
Ref.[38]	99.60%	99.61%	99.61%	99.6067%

Table 11 Comparison of the proposed method and reference methods in terms of UACI of the RGB channels and the average value

Method	R channel	G channel	B channel	Average
Proposed method	33.466%	33.473%	33.447%	33.462%
Ref.[33]	33.4647%	33.5513%	33.4839%	33.5000%
Ref.[3]	33.4567%	33.4759%	33.4578%	33.4634%
Ref.[38]	33.45%	33.43%	33.41%	33.43%

5.6 Medical Image Encryption and Protection Evaluation

With the development of intelligent healthcare systems, remote diagnosis, and cloud-based medical services, medical images have become an important data type in healthcare information transmission and storage. Medical images involve private diagnostic information and patient-related visual data, making secure protection during storage and transmission necessary. Therefore, this section further evaluates the applicability of the proposed encryption scheme to medical image scenarios.

Three representative dermoscopic images, including Melanoma, Nevus, and Basal cell carcinoma, were selected from the publicly available HAM10000 collection in the ISIC Archive for evaluation[39]. During the experiments, the optimized 10-dimensional parameter vector obtained in the previous optimization stage was directly applied to different medical images without additional parameter adjustment. This strategy aims to investigate whether the optimized parameters can maintain stable encryption performance for different image contents.

Fig 11 demonstrates that the ciphertexts generated from different medical images lose recognizable structural patterns while maintaining successful reversibility under the correct parameters. Meanwhile, the decrypted images are completely recovered using the same parameter vector, indicating that the proposed scheme preserves reliable reversibility when applied to different medical images.

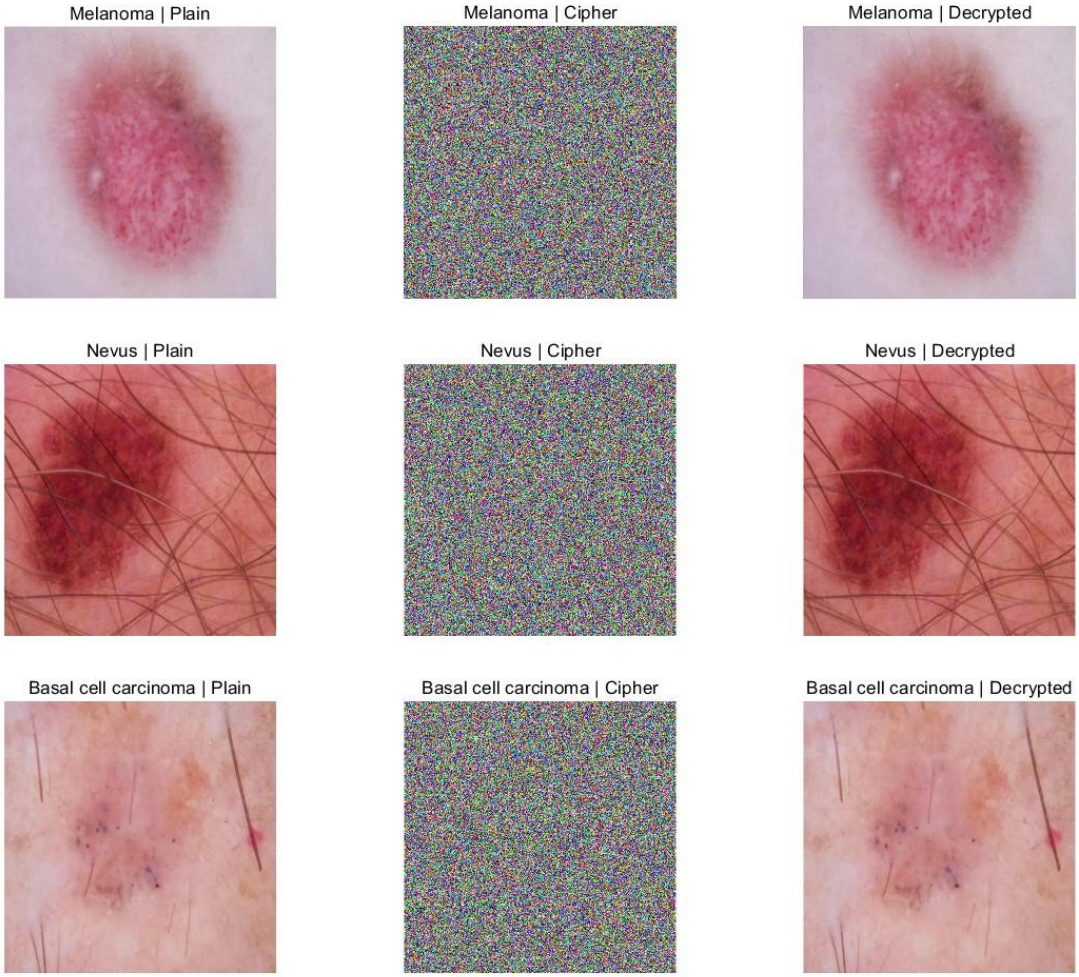


Fig 11 Encryption and decryption results of medical images.

Table 12 presents the security evaluation results of encrypted medical images. The entropy values of all encrypted images are close to the theoretical maximum value of 8, with an average entropy of 7.9971, demonstrating that the proposed scheme maintains desirable statistical characteristics for different medical image contents. Moreover, the mean absolute correlation coefficient between adjacent ciphertext pixels is only 0.0032, indicating that the spatial correlation among ciphertext pixels remains effectively suppressed. For differential sensitivity evaluation, the average NPCR and UACI values of the three medical images reach 99.6056% and 33.4673%, respectively, which are close to the theoretical reference values. These results demonstrate that the proposed encryption scheme maintains strong sensitivity to slight changes in medical image contents.

Table 12 Security performance analysis of encrypted medical images.

Image	Entropy	Mean absolute correlation	NPCR	UACI
Melanoma	7.9969	0.0039	99.6182%	33.5056%
Nevus	7.9972	0.0027	99.5954%	33.4582%
Basal cell carcinoma	7.9971	0.0028	99.6032%	33.4381%
Average	7.9971	0.0032	99.6056%	33.4673%

Overall, the proposed color image encryption method maintains stable encryption performance when extended from conventional test images to medical image scenarios. The above results demonstrate the applicability of the proposed scheme for protecting sensitive medical image data.

6 Conclusion

A color image encryption framework integrating joint parameter optimization with chaotic key-stream reconstruction is presented in this paper. By embedding chaotic-system parameters and reconstruction-related variables into a unified optimization space, the proposed strategy allows the optimized control vector to participate in chaotic sequence generation, RGB synchronous permutation, key-stream rearrangement, and channel-coupled bidirectional diffusion. Extensive evaluations verify that the generated ciphertexts successfully eliminate recognizable image patterns while maintaining desirable statistical characteristics. The security analyses demonstrate that the proposed framework achieves low inter-pixel dependency, strong sensitivity to plaintext variations, and reliable robustness against differential cryptanalysis and common image degradation conditions. Moreover, experiments performed on representative medical images confirm that the optimized parameters can adapt to different image contents while preserving stable encryption performance, indicating its feasibility for protecting sensitive medical image data. Future investigations will concentrate on improving the efficiency of the optimization procedure and developing lightweight versions of the framework for resource-constrained environments. Furthermore, adaptive parameter searching and secure multimedia transmission scenarios will be explored to enhance the practical applicability of the proposed method.

Data Availability

The House and Peppers benchmark images used in this study are publicly available from the USC-SIPI Image Database (https://sipi.usc.edu/database/database.php?volume=misc). The dermoscopic images used for medical image evaluation were selected from the publicly available HAM10000 collection in the ISIC Archive (https://api.isic-archive.com/collections/212/). The Lena image was used as a conventional benchmark image; however, USC-SIPI no longer redistributes this image. The processed numerical data supporting the findings of this study are available from the corresponding author upon reasonable request.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 62201332.

References

1. Amiriebrahimabadi, M., Rouhi, Z. & Mansouri, N. A Comprehensive Survey of Multi-Level Thresholding Segmentation Methods for Image Processing. *Arch. Comput. Methods Eng.* **31**, 3647–3697 (2024).

2. Mahalakshmi, K. & Nagarajan, S. Comprehensive Review and Analysis of Image Encryption Techniques. *IEEE Access* **13**, 109783–109813 (2025).

3. Lai, Q. & Hua, H. A new block-based color image encryption method using 3D hyperchaos. *J. Inf. Secur. Appl.* **99**, 104454 (2026).

4. Song, W. *et al.* Batch image encryption using cross image permutation and diffusion. *J. Inf. Secur. Appl.* **80**, 103686 (2024).

5. Pankaj, S. & Dua, M. Chaos based Medical Image Encryption Techniques: A Comprehensive Review and Analysis. *Inf. Secur. J. Glob. Perspect.* **33**, 332–358 (2024).

6. SaberiKamarposhti, M., Ghorbani, A. & Yadollahi, M. A comprehensive survey on image encryption: Taxonomy, challenges, and future directions. *Chaos Solitons Fractals* **178**, 114361 (2024).

7. Gokcay, E. & Tora, H. A novel data encryption method using an interlaced chaotic transform. *Expert Syst. Appl.* **237**, 121494 (2024).

8. Vijayakumar, M. & Ahilan, A. An optimized chaotic S-box for real-time image encryption scheme based on 4-dimensional memristive hyperchaotic map. *Ain Shams Eng. J.* **15**, 102620 (2024).

9. Rahman, M., Murmu, A., Kumar, P., Moparthi, N. R. & Namasudra, S. A novel compression-based 2D-chaotic sine map for enhancing privacy and security of biometric identification systems. *J. Inf. Secur. Appl.* **80**, 103677 (2024).

10. Zhou, M., Li, X., Du, W., Li, J. & Wei, Z. Pixel-Level and DNA-Level Image Encryption Method

Based on Five-Dimensional Hyperchaotic System. *Entropy* **27**, 1221 (2025).

11. Li, X., Sun, B., Bi, X., Yan, H. & Wang, L. A Novel Color Image Encryption Algorithm Based on Cross-plane Scrambling and Diffusion. *Mob. Netw. Appl.* **29**, 583–594 (2024).

12. Li, L. A novel chaotic map application in image encryption algorithm. *Expert Syst. Appl.* **252**, 124316 (2024).

13. Li, Q. & Wen, Z. Research on Chaos Image Encryption Algorithm Based on Scrambling Diffusion. in *2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* 1036–1039 (IEEE, Chongqing, China, 2024). doi:10.1109/IMCEC59810.2024.10575274.

14. Ponnambalam, M., Ponnambalam, M., Ghazalah, S. A. & Sambas, A. Hybrid inter woven scrambling with spiral shell 3D hyperchaotic diffusion for secure color image encryption. *Nonlinear Dyn.* **113**, 26867–26897 (2025).

15. Li, C., Zhang, Y., Li, H. & Zhou, Y. Visual image encryption scheme based on inter-intra-block scrambling and weighted diffusion. *Vis. Comput.* **40**, 731–746 (2024).

16. Fu, C. *et al.* An efficient and secure medical image protection scheme based on chaotic maps. *Comput. Biol. Med.* **43**, 1000–1010 (2013).

17. Lin, C.-F., Lin, Y.-X. & Chang, S.-H. Medical Image Encryption Using Chaotic Mechanisms: A Study. *Bioengineering* **12**, 734 (2025).

18. Man, Z., Gao, C., Dai, Y. & Meng, X. Dynamic rotation medical image encryption scheme based on improved Lorenz chaos. *Nonlinear Dyn.* **112**, 13571–13597 (2024).

19. Kocak, O., Erkan, U., Toktas, A. & Gao, S. PSO-based image encryption scheme using modular integrated logistic exponential map. *Expert Syst. Appl.* **237**, 121452 (2024).

20. Al Jawher, W. A. M. & Abdul Kareem, A. A. An Image Encryption Using Hybrid Grey Wolf Optimization and Chaotic Map. *Int. J. Inf. Comput. Secur.* **1**, 10057701 (2023).

21. Qin, X. & Zhang, Y. Image encryption algorithm based on COA and hyperchaotic Lorenz system. *Nonlinear Dyn.* **112**, 10611–10632 (2024).

22. Lai, Q. & Liu, Y. A cross-channel color image encryption algorithm using two-dimensional hyperchaotic map. *Expert Syst. Appl.* **223**, 119923 (2023).

23. Toktas, F., Erkan, U. & Yetgin, Z. Cross-channel color image encryption through 2D hyperchaotic hybrid map of optimization test functions. *Expert Syst. Appl.* **249**, 123583 (2024).

24. Wen, H. & Lin, Y. Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding. *Expert Syst. Appl.* **237**, 121514 (2024).

25. Yin, Z., Abel, A., Tang, J., Zhang, X. & Luo, B. Reversible data hiding in encrypted images based on multi-level encryption and block histogram modification. *Multimed. Tools Appl.* **76**, 3899–3920 (2017).

26. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **37**, 145–151 (1991).

27. Zhang, Y. A new unified image encryption algorithm based on a lifting transformation and chaos. *Inf. Sci.* **547**, 307–327 (2021).

28. Alghamdi, Y. & Munir, A. Image Encryption Algorithms: A Survey of Design and Evaluation Metrics. *J. Cybersecurity Priv.* **4**, 126–152 (2024).

29. Wei, D., Jiang, M. & Deng, Y. A secure image encryption algorithm based on hyper-chaotic and bit-level permutation. *Expert Syst. Appl.* **213**, 119074 (2023).

30. Ye, G., Pan, C., Huang, X., Zhao, Z. & He, J. A Chaotic Image Encryption Algorithm Based on Information Entropy. *Int. J. Bifurc. Chaos* **28**, 1850010 (2018).

31. Huang, X., Dong, Y., Ye, G., Yap, W.-S. & Goi, B.-M. Visually meaningful image encryption algorithm based on digital signature. *Digit. Commun. Netw.* **9**, 159–165 (2023).

32. Yu, J., Xie, W., Zhong, Z. & Wang, H. Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation. *Chaos Solitons Fractals* **162**, 112456 (2022).

33. Chu, R. & Zhang, S. A color image encryption based on chaotic system and PSO-BP neural network and DNA mutation. *Phys. Scr.* **97**, 125216 (2022).

34. Li, S., Yi, Y., Chen, Z. & Xie, S. A hybrid color image encryption scheme integrating 3D hyperchaotic map, complex network, and compressed sensing. *J. Inf. Secur. Appl.* **99**, 104441 (2026).

35. Abed, Q. K. & Al-Jawher, W. A. M. New Colorful Image Encryption Method Using Triple Chaotic Maps and Grey Wolf Optimization (GWO). *J. Port Sci. Res.* **7**, 228–245 (2024).

36. Minocha, S., Sharma, S. R., Singh, B. & Gandomi, A. H. Adaptive image encryption approach using an enhanced swarm intelligence algorithm. *Sci. Rep.* **15**, 9476 (2025).

37. Luo, Y., Ouyang, X., Liu, J., Cao, L. & Zou, Y. An image encryption scheme based on particle swarm optimization algorithm and hyperchaotic system. *Soft Comput.* **26**, 5409–5435 (2022).

38. Yang, F., Mou, J., Sun, K., Cao, Y. & Jin, J. Color Image Compression-Encryption Algorithm Based on Fractional-Order Memristor Chaotic Circuit. *IEEE Access* **7**, 58751–58763 (2019).

39. Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data* **5**, 180161 (2018).