Networks and
Heterogeneous Media

http://www.aimspress.com/journal/nhm

NHM, 18(3): 1260–1287.
DOI: 10.3934/nhm.2023055
Received: 03 January 2023
Revised: 08 March 2023
Accepted: 27 March 2023
Published: 25 April 2023

Research article

An image encryption algorithm based on heat flow cryptosystems

Jin Li[1,*], **Jinzheng Qu**[1], **Xibo Duan**[2] **and Xiaoning Su**[1]

[1] College of Science, North China University of Science and Technology, Hebei Key Laboratory of Data Science and Application, Tangshan 063210, P. R. China

[2] Shandong Water Conservancy Vocational College, Rizhao 276800, P. R. China

* **Correspondence:** Email: lijin@lsec.cc.ac.cn; Tel: +86-186-1518-6091.

Abstract: Image encryption has been an important research topic in information security. Different from traditional encryption methods, heat flow cryptosystem is a new encryption method. This paper proposes an image encryption algorithm based on heat flow cryptosystem. First, a class of heat flow cryptosystem based on nonlinear pseudo-parabolic equations are given in this paper. Second, a numerical method with high precision namely barycentric Lagrange interpolation collocation method is proposed to solve the nonlinear pseudo-parabolic equation. Third, an image encryption algorithm based on the heat flow cryptosystem is designed, the detailed process of encryption and decryption algorithm is given, the flow diagram of algorithm is showed. Finally, the proposed encryption algorithm is applied to various image with gray and RGB format and compared with the current popular chaotic encryption algorithm. Many indicators such as histograms, information entropy and correlation are used to objectively evaluate the image encryption algorithm. The experimental results show that the proposed image encryption algorithm is better in most indicators and the algorithm is sensitive to the change of key and plaintext.

Keywords: information security; image encryption; heat flow cryptosystem; barycentric Lagrange interpolation

1. Introduction

The development of information technology leads to information security issues that have become a serious problem, so it is particularly important to develop the encryption technology. For example, traditional symmetric encryption has DES and AES encryption [1] etc., asymmetric encryption has RSA encryption [1] etc. Due to the large amount of data for image, the traditional encryption technology for image encryption effect is not prominent. Therefore, many encryption algorithms are proposed and improved, chaotic encryption system is the most widely used encryption method in recent years [2, 3], such as the improved Henon map [4], 3D logistic-sine cascade map [5] and improved sine map [6]. In [7], a new chaotic system called delayed feedback dynamic mixed linear-nonlinear coupled mapping lattices (DFDMLNCML) is proposed, in [8], an image encryption algorithm based on a hidden attractor chaos system and Knuth-Durstenfeld algorithm is proposed, in [9], a chaos encryption algorithm combining Cyclic Redundancy Check (CRC) and nine palace map is proposed and in [10], a chaotic image encryption algorithm combined with frequency-domain DNA encoding is proposed. The authors of [11] applied speech recognition technology to chaotic cryptosystems. An image encryption algorithm based on dynamic row scrambling and Zigzag transformation is proposed in [12], the authors of [13] proposed a new chaotic pseudo-random number generator (CPRNG). A colour image encryption model based on two nearby orbits of chaotic systems is presented in [14]. At present, chaotic systems have also been widely used in various practical problems, such as medical images [15–18], secure wireless communication [19], IoT applications [20, 21] and so on. In addition, many other encryption algorithms have been proposed in recent years, such as the image encryption algorithm based on self-adaptive diffusion and combined global scrambling [22], the image encryption algorithm based on 2-dimensional compressive sensing [23], the fast image encryption algorithm based on parallel computing system [24]. With the continuous development of encryption technology, new encryption algorithms are emerging. Therefore, the evaluation of encryption algorithms is more stringent, and some evaluation indexes are used to test the security and randomness of images, such as Statistical Test NIST SP 800-22 [25], Test U01 [26] and the 0–1 test [27] etc. Encryption technology is still in continuous development. Due to the emergence of various means of attack, the new encryption algorithm must and deserves to be proposed.

Different from traditional encryption technique, the heat flow cryptosystem is a new method that combines the cryptographic method with the partial differential equation [28], the core of the cryptosystem is a class of partial differential equation. The encryption and decryption process is realized by solving the equation to get the state function values at two moments, thus it is usually necessary to seek efficient and stable numerical methods to solve such equations. For the research of heat flow cryptosystem, many scholars have proposed the reliable numerical methods to solve the models, such as reproducing kernel method [29], pseudo-spectral method [30], finite element method [31], mixed volume element method [32], difference method [33] and so on. In addition, the model of heat flow cryptosystem is also applied to practical problems, such as image encryption [34], text and continuous signals encryption. In this paper, the barycentric Lagrange interpolation collocation method is adopted for heat flow cryptosystem. Barycentric interpolation collocation method is a meshless numerical method with high computation precision and efficiency [35], thus the numerical method has been widely used [36].

The main work of this paper is to propose an image encryption algorithm with high security, and then we use several common indicators to objectively evaluate the encryption algorithm. Block encryption is adopted and each group contains two diffusions and one scrambling process. The first diffusion changes the pixel value of the image through the nonlinear pseudo-parabolic equation. Further, each group is recombined into a matrix, matrix elements represent the pixel values after diffusion. The second diffusion performs bitwise XOR on each group and its previous group and we scramble the matrix, then each group performs the same operation. The encryption algorithm is sensitive to the pixel value of the first group. If a pixel value in the first group is changed, it will affect the diffusion effect of the subsequent groups. Therefore, the algorithm has strong key sensitivity and plaintext sensitivity. Compared with the common chaotic encryption system, the advantage of the heat flow cryptosystem is that the key space is so large, the key functions $p(x, y)$ and $q(x, y)$ can be any continuous function in the domain, so the selected $p(x, y)$ and $q(x, y)$ are infinite. Heat flow cryptosystem can effectively resist exhaustive attack by relying on this ability.

The structure of this article is organized as follows: in Section 2, the heat flow cryptosystem based on a class of (1+1)-dimensional and (2+1)-dimensional nonlinear pseudo-parabolic equations are given, the barycentric Lagrange interpolation to solve the nonlinear pseudo-parabolic equations is introduced. In Section 3, the image encryption algorithm based on this heat flow cryptosystem is designed, the flow diagram is also given. In Section 4, the decryption algorithm and flow diagram is given. In Section 5, several examples are given to support the proposed encryption algorithm.

2. Barycentric Lagrange interpolation collocation method for heat flow cryptosystem

2.1. The encryption and decryption model of heat flow cryptosystem

In this section, the model of heat flow cryptosystem based on a (1+1)-dimensional nonlinear pseudo-parabolic equation is given [31]

$$\begin{cases}(p(x)u_{xt})_x + (p(x)u_x)_x - (q(x) + 1)u_t - q(x)u = f(x, t, r) + 20xu^2,\\ u(x, 0) = f_0(x),\\ u(0, t) = 0, u(1, t) = 0\end{cases} \tag{2.1}$$

and the model of heat flow cryptosystem based on (2+1)-dimensional nonlinear pseudo-parabolic equation is given [32]

$$\begin{cases}\nabla \cdot (p(x, y)\nabla u_t) + \nabla \cdot (p(x, y)\nabla u) - (q(x, y) - 1)u_t - q(x, y)u = f(x, y, t, r) + 20xyu^2,\\ u(x, y, 0) = f_0(x, y),\\ u(0, y, t) = 0, u(1, y, t) = 0, u(x, 0, t) = 0, u(x, 1, t) = 0,\end{cases} \tag{2.2}$$

where $(x, t) \in [0, 1] \times [0, T]$ for (1+1)-dimensional nonlinear pseudo-parabolic equation (2.1) and $(x, y) \in \Omega = [0, 1] \times [0, 1]$, $t \in [0, T]$ for (2+1)-dimensional nonlinear pseudo-parabolic equation Eq (2.2), T is the final time. $p(x)$, $q(x)$ in (2.1) and $p(x, y)$, $q(x, y)$ in Eq (2.2) are given functions, ∇ is Nabla operator and $(\nabla\cdot)$ is divergence operator. The r in formula (2.1) and (2.2) is a constant.

Let

$$\begin{cases}S_1[u] = (p(x)u_{xt})_x + (p(x)u_x)_x - (q(x) + 1)u_t - q(x)u,\\ S_2[u] = \nabla \cdot (p(x, y)\nabla u_t) + \nabla \cdot (p(x, y)\nabla u) - (q(x, y) - 1)u_t - q(x, y)u.\end{cases}$$

We take the function values at $t = 0$ (i.e., $f_0(x)$ and $f_0(x, y)$) as plaintext and take $u(x, T) = g_0(x)$, $u(x, y, T) = g_0(x, y)$ as ciphertext, then the (1+1)-dimensional encryption model and decryption model can be expressed as

$$\begin{cases}S_1[u] = f(x, t, r) + 20xu^2,\\ u(x, 0) = f_0(x),\\ u(0, t) = 0, u(1, t) = 0,\end{cases} \quad \begin{cases}S_1[u] = f(x, t, r) + 20xu^2,\\ u(x, T) = g_0(x),\\ u(0, t) = 0, u(1, t) = 0,\end{cases} \tag{2.3}$$

(2+1)-dimensional encryption model and decryption model can be expressed as

$$\begin{cases} S_2[u] = f(x,y,t,r) + 20xyu^2, \\ u(x,y,0) = f_0(x,y), \\ u(0,y,t) = 0, u(1,y,t) = 0, \\ u(x,0,t) = 0, u(x,1,t) = 0, \end{cases} \quad \begin{cases} S_2[u] = f(x,y,t,r) + 20xyu^2, \\ u(x,y,T) = g_0(x,y), \\ u(0,y,t) = 0, u(1,y,t) = 0, \\ u(x,0,t) = 0, u(x,1,t) = 0. \end{cases} \tag{2.4}$$

The $p(x), q(x), p(x,y), q(x,y), f(x,t,r), f(x,y,t,r)$ and final time T are cryptographic key. The plaintext and ciphertext corresponding to the solution of u at two different times $t = 0$ and $t = T$.

2.2. Iterative scheme for nonlinear pseudo-parabolic equation

The (2+1)-dimensional nonlinear pseudo-parabolic equation (2.2) is taken as an example to construct the iterative scheme. We can obtain the simplified equation (2.2)

$$(p_x + p_y)(u_{xt} + u_{yt} + u_x + u_y) + p(u_{xxt} + u_{yyt} + u_{xx} + u_{yy}) - (q+1)u_t - qu = f(x,y,t,r) + 20xyu^2. \tag{2.5}$$

Substituting the iterative initial value u_0 into the nonlinear term of $20xyu^2$, the linearized equation can be obtained as

$$(p_x + p_y)(u_{xt} + u_{yt} + u_x + u_y) + p(u_{xxt} + u_{yyt} + u_{xx} + u_{yy}) - (q+1)u_t - qu = f(x,y,t,r) + 20xyu_0^2. \tag{2.6}$$

The iterative scheme can be constructed by $u \to u^{(h)}$, $u_0 \to u^{(h-1)}$

$$(p_x+p_y)(u_{xt}^{(h)}+u_{yt}^{(h)}+u_x^{(h)}+u_y^{(h)})+p(u_{xxt}^{(h)}+u_{yyt}^{(h)}+u_{xx}^{(h)}+u_{yy}^{(h)})-(q+1)u_t^{(h)}-qu^{(h)} = f(x,y,t,r)+20xy(u^{(h-1)})^2. \tag{2.7}$$

2.3. Collocation scheme for iterative scheme

Consider $u(x,y,t) \in \Omega = [a,b] \times [c,d]$, the spatial region Ω is discretized into

$$\left\{\Omega_{ij} = (x_i, y_j), i = 1, 2, \cdots, m; j = 1, 2, \cdots, n\right\}$$

and time region $[0,T]$ is discretized into $\{t_k, k = 1, 2, \cdots, l\}$, Then we have tensor interpolation nodes

$$\left\{(x_i, y_j, t_k), i = 1, 2, \cdots, m; j = 1, 2, \cdots, n; k = 1, 2, \cdots, l\right\} \tag{2.8}$$

in the computational domain $\Omega \times [0,T]$. Defining $u_{ijk} = u(x_i, y_j, t_k)$, then the barycentric Lagrange interpolation polynomial of $u(x,y,t)$ can be expressed as

$$u(x,y,t) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} L_i(x) M_j(y) T_k(t) u_{ijk}, \tag{2.9}$$

where $L_i(x)$, $M_j(y)$, $T_k(t)$ are the basic function, the corresponding expressions are

$$\begin{cases} L_i(x) = \dfrac{\dfrac{\omega_i}{x-x_i}}{\displaystyle\sum_{i=1}^{m}\dfrac{\omega_i}{x-x_i}}, \\ \omega_i = \dfrac{1}{\displaystyle\prod_{k=1,k\neq i}^{m}(x_i-x_k)}, \end{cases} \quad \begin{cases} M_j(y) = \dfrac{\dfrac{\nu_j}{y-y_j}}{\displaystyle\sum_{j=1}^{n}\dfrac{\nu_j}{y-y_j}}, \\ \nu_j = \dfrac{1}{\displaystyle\prod_{k=1,k\neq j}^{n}(y_j-y_k)}, \end{cases} \quad \begin{cases} T_k(t) = \dfrac{\dfrac{\lambda_k}{t-t_k}}{\displaystyle\sum_{k=1}^{l}\dfrac{\lambda_k}{t-t_k}}, \\ \lambda_k = \dfrac{1}{\displaystyle\prod_{i=1,i\neq k}^{l}(t_k-t_i)}. \end{cases}$$

Adopting Eq (2.9) to approximate the unknown function u in Eq (2.7) and make it hold at the interpolation node (2.8), then the iterative scheme (2.7) can be written in the matrix form (Ref. [37])

$$\boldsymbol{H}\boldsymbol{u}^{(h)} = \boldsymbol{f} + 20\boldsymbol{x}\boldsymbol{y}(\boldsymbol{u}^{(h-1)})^2, \tag{2.10}$$

where

$$\begin{aligned} \boldsymbol{H} = {} & (\boldsymbol{P}_x + \boldsymbol{P}_y)\left[(\boldsymbol{L}^{(100)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{L}^{(001)}) + (\boldsymbol{I}_m \otimes \boldsymbol{L}^{(010)} \otimes \boldsymbol{L}^{(001)}) + (\boldsymbol{L}^{(100)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l) + (\boldsymbol{I}_m \otimes \boldsymbol{L}^{(010)} \otimes \boldsymbol{I}_l)\right] \\ & + \boldsymbol{P}\left[(\boldsymbol{L}^{(200)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{L}^{(001)}) + (\boldsymbol{I}_m \otimes \boldsymbol{L}^{(020)} \otimes \boldsymbol{L}^{(001)}) + (\boldsymbol{L}^{(200)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l) + (\boldsymbol{I}_m \otimes \boldsymbol{L}^{(020)} \otimes \boldsymbol{I}_l)\right] \\ & - (\boldsymbol{Q} + \boldsymbol{I}_m \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l)(\boldsymbol{I}_m \otimes \boldsymbol{I}_n \otimes \boldsymbol{L}^{(001)}) - \boldsymbol{Q}(\boldsymbol{I}_m \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l), \end{aligned}$$

$$\boldsymbol{P}_x = \operatorname{diag}(\boldsymbol{p}_x)\otimes\boldsymbol{I}_l, \boldsymbol{p}_x = [p_x(x_1,y_1), p_x(x_1,y_2), \cdots p_x(x_1,y_n), \cdots, \cdots, p_x(x_m,y_1), p_x(x_m,y_2), \cdots p_x(x_m,y_n)]^{\mathrm{T}},$$

$$\boldsymbol{P}_y = \operatorname{diag}(\boldsymbol{p}_y)\otimes\boldsymbol{I}_l, \boldsymbol{p}_y = \left[p_y(x_1,y_1), p_y(x_1,y_2), \cdots p_y(x_1,y_n), \cdots, \cdots, p_y(x_m,y_1), p_y(x_m,y_2), \cdots p_y(x_m,y_n)\right]^{\mathrm{T}},$$

$$\boldsymbol{P} = \operatorname{diag}(\boldsymbol{p}) \otimes \boldsymbol{I}_l, \boldsymbol{p} = [p(x_1,y_1), p(x_1,y_2), \cdots p(x_1,y_n), \cdots, \cdots, p(x_m,y_1), p(x_m,y_2), \cdots p(x_m,y_n)]^{\mathrm{T}},$$

$$\boldsymbol{Q} = \operatorname{diag}(\boldsymbol{q}) \otimes \boldsymbol{I}_l, \boldsymbol{q} = [q(x_1,y_1), q(x_1,y_2), \cdots q(x_1,y_n), \cdots, \cdots, q(x_m,y_1), q(x_m,y_2), \cdots q(x_m,y_n)]^{\mathrm{T}},$$

$$\boldsymbol{f} = [f_{111}, f_{112}, \cdots, f_{11l}, f_{121}, f_{122}, \cdots, f_{12n}, \cdots, \cdots, f_{mn1}, f_{mn2}, \cdots, f_{mnl}]^{\mathrm{T}}, f_{ijk} = f(x_i, y_j, t_k),$$

$$\boldsymbol{u}^{(h)} = \left[u_{111}^{(h)}, u_{112}^{(h)}, \cdots, u_{11l}^{(h)}, u_{121}^{(h)}, u_{122}^{(h)}, \cdots, u_{12l}^{(h)}, \cdots, \cdots, u_{mn1}^{(h)}, u_{mn2}^{(h)}, \cdots, u_{mnl}^{(h)}\right]^{\mathrm{T}}, u_{ijk}^{(h)} = u^{(h)}(x_i, y_j, t_k),$$

$$\boldsymbol{u}^{(h-1)} = \left[u_{111}^{(h-1)}, u_{112}^{(h-1)}, \cdots, u_{11l}^{(h-1)}, u_{121}^{(h-1)}, u_{122}^{(h-1)}, \cdots, u_{12l}^{(h-1)}, \cdots, \cdots, u_{mn1}^{(h-1)}, u_{mn2}^{(h-1)}, \cdots, u_{mnl}^{(h-1)}\right]^{\mathrm{T}},$$

$$u_{ijk}^{(h-1)} = u^{(h-1)}(x_i, y_j, t_k),$$

where $\bigotimes$ is kronecker product, $\boldsymbol{I}_m$, $\boldsymbol{I}_n$, $\boldsymbol{I}_l$ are unit matrices with m-order, n-order, l-order, respectively. $\boldsymbol{L}^{(100)}$ and $\boldsymbol{L}^{(200)}$ are the 1-order and 2-order differential matrices with x. $\boldsymbol{L}^{(010)}$ and $\boldsymbol{L}^{(020)}$ are 1-order and 2-order differential matrices with y. $\boldsymbol{L}^{(001)}$ is the 1-order differential matrices with t. The elements in differential matrix and the handling of initial-boundary condition are given in Ref. [37].

Let $h = 1$, the matrix equation (2.10) begins iteration. The iterated precision E is given, when

$$\|\boldsymbol{u}^{(h)} - \boldsymbol{u}^{(h-1)}\|_2 = \sqrt{\begin{aligned} & (u_{111}^{(h)} - u_{111}^{(h-1)})^2 + (u_{112}^{(h)} - u_{112}^{(h-1)})^2 + \cdots + (u_{11l}^{(h)} - u_{11l}^{(h-1)})^2 \\ & + (u_{121}^{(h)} - u_{121}^{(h-1)})^2 + (u_{122}^{(h)} - u_{122}^{(h-1)})^2 + \cdots + (u_{12l}^{(h)} - u_{12l}^{(h-1)})^2 \\ & + \cdots + \cdots \\ & + (u_{mn1}^{(h)} - u_{mn1}^{(h-1)})^2 + (u_{mn2}^{(h)} - u_{mn2}^{(h-1)})^2 + \cdots + (u_{mnl}^{(h)} - u_{mnl}^{(h-1)})^2 \end{aligned}} \leq E, \tag{2.11}$$

iteration stop, we denote $h = h_{\text{end}}$ for the last iteration, the numerical solutions $u_{ijk}^{(h_{\text{end}})} = u^{(h_{\text{end}})}(x_i, y_j, t_k), i = 1, 2, \cdots, m; j = 1, 2, \cdots, n; k = 1, 2, \cdots, l$ can be obtained. When the $k = 1$, the

$$\boldsymbol{u}_{t=0}^{(h_{\text{end}})} = \left[u_{111}^{(h_{\text{end}})}, u_{121}^{(h_{\text{end}})}, \cdots, u_{1n1}^{(h_{\text{end}})}, u_{211}^{(h_{\text{end}})}, u_{221}^{(h_{\text{end}})}, \cdots, u_{2n1}^{(h_{\text{end}})}, \cdots, \cdots, u_{m11}^{(h_{\text{end}})}, u_{m21}^{(h_{\text{end}})}, \cdots, u_{mn1}^{(h_{\text{end}})}\right]^{\mathrm{T}} \quad (2.12)$$

is the numerical solutions at $t = 0$.

When the $k = l$, the

$$\boldsymbol{u}_{t=T}^{(h_{\text{end}})} = \left[u_{11l}^{(h_{\text{end}})}, u_{12l}^{(h_{\text{end}})}, \cdots, u_{1nl}^{(h_{\text{end}})}, u_{21l}^{(h_{\text{end}})}, u_{22l}^{(h_{\text{end}})}, \cdots, u_{2nl}^{(h_{\text{end}})}, \cdots, \cdots, u_{m1l}^{(h_{\text{end}})}, u_{m2l}^{(h_{\text{end}})}, \cdots, u_{mnl}^{(h_{\text{end}})}\right]^{\mathrm{T}} \quad (2.13)$$

is the numerical solutions at $t = T$.

3. Image encryption algorithm design

In this section, an image encryption algorithm based on the above analysis is proposed.

Step 1 Consider an image A of size $M \times N$ (Height is M and Width is N), we expand pixels in column order to get a set of 1-dimensional sequences

$$A = \{a_{11}, a_{21}, \cdots, a_{M1}, a_{12}, a_{22}, \cdots, a_{M2}, \cdots, \cdots, a_{1N}, a_{2N}, \cdots, a_{MN}\},$$

where $a_{ij}(i = 1, 2, \cdots, M; j = 1, 2, \cdots, N)$ is the pixel value of row i, column j in the image.

Step 2 Spliting A in order, 16 pixels as a group, we can obtain $B = \{b_1, b_2, \cdots, b_S\}$, where $b_s(s = 1, 2, \cdots, S)$ denote the group s. If the number of pixels in the last group b_S is less than 16, we fill it with 0 pixels.

Step 3 Each group $b_s(s = 1, 2, \cdots, S)$ is normalized, according to the distribution of pixels, the following three cases will be considered:

- If all pixel values are 0 in this group, we adopt the median 0.5 of $[0, 1]$ as the result of pixel normalization.
- If the pixel values are all equal and not 0 in this group, we use the weight of the pixel value in $[0, 255]$ as the normalized result. For example, if the pixel values in this group are all 241, then the normalized results are all $241/255 = 0.9451$.
- If the pixel values are not all equal in this group, we use the formula

$$b_{sk}^{'} = \frac{b_{sk} - b_{s\min}}{b_{s\max} - b_{s\min}} \quad (3.1)$$

where $b_{sk}(s = 1, 2, \cdots, S; k = 1, 2, \cdots, 16)$ denote the k-th pixel value in the group s, $b_{s\max}$ and $b_{s\min}$ denote the maximum and minimum pixel value in the group s, respectively, $b_{sk}^{'}$ is the normalized result.

Then $B^{'} = \left\{b_1^{'}, b_2^{'}, \cdots, b_S^{'}\right\}$ can be obtained. Taking $b_s^{'}$ as the initial condition f_0 in model (2.3) or (2.4). The barycentric Lagrange interpolation collocation method introduced in Section 2 is adopted to solve the nonlinear pseudo-parabolic equation in model (2.3) or (2.4), then we can obtain the g_0 in (2.3) or (2.4), we denote it by g_s.

In the process, $r \in [r_{\min}, r_{\max}]$ is a key parameter, where $r_{\max}$ is a given value and $r_{\min}$ is obtained by the original image

$$\begin{cases} A(i) = A(i,1) \oplus A(i,2) \oplus \cdots \oplus A(i,N); \quad i = 1,2,\cdots,M, \\ E = A(1) \oplus A(2) \oplus \cdots \oplus A(M), \\ r_{\min} = 25 + \text{mod}(E,5), \end{cases}$$

where $\oplus$ denote XOR, $A(i,j)(i = 1,2,\cdots,M; j = 1,2,\cdots,N)$ denote the pixel of position (i,j) in original image, the mod() is the modular function. Each group $b_s^{'}$ corresponds to an r. The growth step of r denote r_h, the r for $b_s^{'}$ is

$$r_s = r_{\min} + (s-1)r_h, s = 1,2,\cdots,S,$$

in addition, a constraint condition is given: when $r \geq r_{\max}$, r start again from $r_{\min}$ according to the same rule. The value of r in the last group $b_S^{'}$ is recorded as r_{end}. We can get the $\{g_1, g_2, \cdots, g_S\}$ and g_s is normalized by use formula (3.1) to get $g_s^{'}$, then we have $C = \{c_1, c_2, \cdots, c_S\}$ by $c_s = g_s^{'} \times 255$.

Step 4 For the $c_s \in C$ in *Step 3*, we combine them in column order into a matrix of size 4 × 4 ,we denote it into $\boldsymbol{c}_s$ as follows

$$\boldsymbol{c}_s = \begin{pmatrix} c_{s(1)} & c_{s(5)} & c_{s(9)} & c_{s(13)} \\ c_{s(2)} & c_{s(6)} & c_{s(10)} & c_{s(14)} \\ c_{s(3)} & c_{s(7)} & c_{s(11)} & c_{s(15)} \\ c_{s(4)} & c_{s(8)} & c_{s(12)} & c_{s(16)} \end{pmatrix}, s = 1,2,\cdots,S,$$

where $c_{s(k)}(k = 1,2,\cdots,16)$ denote the k-th pixel value in c_s. The matrix sequence $\boldsymbol{D}$ formed by $\boldsymbol{d}_s$ is denoted as $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_S\}$.

Step 5 For the $\boldsymbol{c}_1 \in \boldsymbol{C}$ in *Step* 4,

$$\overline{\boldsymbol{c}}_1 = \boldsymbol{c}_1 \oplus \boldsymbol{c}_S = \begin{pmatrix} c_{1(1)} & c_{1(5)} & c_{1(9)} & c_{1(13)} \\ c_{1(2)} & c_{1(6)} & c_{1(10)} & c_{1(14)} \\ c_{1(3)} & c_{1(7)} & c_{1(11)} & c_{1(15)} \\ c_{1(4)} & c_{1(8)} & c_{1(12)} & c_{1(16)} \end{pmatrix} \oplus \begin{pmatrix} c_{S(1)} & c_{S(5)} & c_{S(9)} & c_{S(13)} \\ c_{S(2)} & c_{S(6)} & c_{S(10)} & c_{S(14)} \\ c_{S(3)} & c_{S(7)} & c_{S(11)} & c_{S(15)} \\ c_{S(4)} & c_{S(8)} & c_{S(12)} & c_{S(16)} \end{pmatrix} = \begin{pmatrix} \overline{c}_{1(1)} & \overline{c}_{1(5)} & \overline{c}_{1(9)} & \overline{c}_{1(13)} \\ \overline{c}_{1(2)} & \overline{c}_{1(6)} & \overline{c}_{1(10)} & \overline{c}_{1(14)} \\ \overline{c}_{1(3)} & \overline{c}_{1(7)} & \overline{c}_{1(11)} & \overline{c}_{1(15)} \\ \overline{c}_{1(4)} & \overline{c}_{1(8)} & \overline{c}_{1(12)} & \overline{c}_{1(16)} \end{pmatrix},$$

then we scramble $\overline{\boldsymbol{c}}_1$ according to the rule: the row i moves one position to the left and the column i moves one position up, where $i = 1,2,\cdots,4$, we have the scrambled matrix $\widehat{\boldsymbol{c}_1}$

$$\widehat{\boldsymbol{c}_1} = \begin{pmatrix} \overline{c}_{1(2)} & \overline{c}_{1(10)} & \overline{c}_{1(14)} & \overline{c}_{1(3)} \\ \overline{c}_{1(6)} & \overline{c}_{1(7)} & \overline{c}_{1(15)} & \overline{c}_{1(4)} \\ \overline{c}_{1(8)} & \overline{c}_{1(11)} & \overline{c}_{1(12)} & \overline{c}_{1(5)} \\ \overline{c}_{1(9)} & \overline{c}_{1(13)} & \overline{c}_{1(16)} & \overline{c}_{1(1)} \end{pmatrix},$$

$\overline{\boldsymbol{c}}_2$ can be obtained by

$$\overline{\boldsymbol{c}}_2 = \boldsymbol{c}_2 \oplus \widehat{\boldsymbol{c}_1} = \begin{pmatrix} c_{2(1)} & c_{2(5)} & c_{2(9)} & c_{2(13)} \\ c_{2(2)} & c_{2(6)} & c_{2(10)} & c_{2(14)} \\ c_{2(3)} & c_{2(7)} & c_{2(11)} & c_{2(15)} \\ c_{2(4)} & c_{2(8)} & c_{2(12)} & c_{2(16)} \end{pmatrix} \oplus \begin{pmatrix} \overline{c}_{1(2)} & \overline{c}_{1(10)} & \overline{c}_{1(14)} & \overline{c}_{1(3)} \\ \overline{c}_{1(6)} & \overline{c}_{1(7)} & \overline{c}_{1(15)} & \overline{c}_{1(4)} \\ \overline{c}_{1(8)} & \overline{c}_{1(11)} & \overline{c}_{1(12)} & \overline{c}_{1(5)} \\ \overline{c}_{1(9)} & \overline{c}_{1(13)} & \overline{c}_{1(16)} & \overline{c}_{1(1)} \end{pmatrix} = \begin{pmatrix} \overline{c}_{2(1)} & \overline{c}_{2(5)} & \overline{c}_{2(9)} & \overline{c}_{2(13)} \\ \overline{c}_{2(2)} & \overline{c}_{2(6)} & \overline{c}_{2(10)} & \overline{c}_{2(14)} \\ \overline{c}_{2(3)} & \overline{c}_{2(7)} & \overline{c}_{2(11)} & \overline{c}_{2(15)} \\ \overline{c}_{2(4)} & \overline{c}_{2(8)} & \overline{c}_{2(12)} & \overline{c}_{2(16)} \end{pmatrix},$$

we scramble $\overline{\boldsymbol{c}}_2$ according to the rule: the row j moves one position to the right and the column j moves one position down, where $j = 1, 2, \cdots, 4$, we have the scrambled matrix $\widetilde{\boldsymbol{c}}_2$

$$\widetilde{\boldsymbol{c}}_2 = \begin{pmatrix} \overline{c}_{2(4)} & \overline{c}_{2(8)} & \overline{c}_{2(12)} & \overline{c}_{2(13)} \\ \overline{c}_{2(14)} & \overline{c}_{2(1)} & \overline{c}_{2(5)} & \overline{c}_{2(9)} \\ \overline{c}_{2(15)} & \overline{c}_{2(2)} & \overline{c}_{2(6)} & \overline{c}_{2(10)} \\ \overline{c}_{2(16)} & \overline{c}_{2(3)} & \overline{c}_{2(7)} & \overline{c}_{2(11)} \end{pmatrix},$$

Starting from $\boldsymbol{c}_3$,

$$\begin{cases} \overline{\boldsymbol{c}}_s = \boldsymbol{c}_s \oplus \widetilde{\boldsymbol{c}}_{s-1}, \overline{\boldsymbol{c}}_s \to \widehat{\boldsymbol{c}}_s & \mathrm{mod}(s, 2) \neq 0, s \geq 3 \\ \overline{\boldsymbol{c}}_s = \boldsymbol{c}_s \oplus \widehat{\boldsymbol{c}}_{s-1}, \overline{\boldsymbol{c}}_s \to \widetilde{\boldsymbol{c}}_s, & \mathrm{mod}(s, 2) = 0, s \geq 3 \end{cases}$$

then the $\boldsymbol{C} = \{\widehat{\boldsymbol{c}}_1, \widetilde{\boldsymbol{c}}_2, \widehat{\boldsymbol{c}}_3, \widetilde{\boldsymbol{c}}_4, \cdots\}$ can be obtained. The diagram for *Step* 5 is shown in Figure 1.

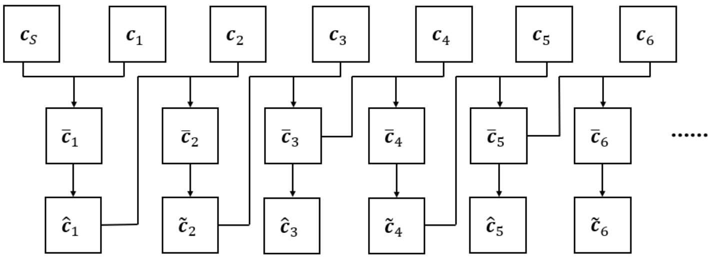


Figure 1. The diagram of *Step* 5.

Step 6 For the each matrix of $\boldsymbol{C}$ in *Step* 5, we expand it in column order, then a long 1-dimensional sequence can be obtained. Extracting the first MN elements and combine it into a matrix of size $M \times N$ in column order, then we can get the encrypted image.

In Algorithm 1, we give the pseudo-code of Image preprocessing process corresponding to the Step 1 to Step 2. The Algorithm 2 shows the process of heat flow cryptosystem encryption in Step 3.

4. Image decryption algorithm design

In heat flow cryptosystem, the function value at final time can be obtained by taking the plaintext as the initial condition. The reverse process is to solve the function value at initial time by taking the final time function value as the initial condition. In addition, each step of encryption algorithm is reversible, so the decryption algorithm is the inverse process of encryption algorithm. The specific decryption algorithm process as follows.

Step 1 For the 1-dimensional sequence $\boldsymbol{C}$ in encryption algorithm *Step* 6, supposing the length of $\boldsymbol{C}$ is l_C, recombining the $\boldsymbol{C}$ in column order into a matrix of size $4 \times (l_C/4)$.

Step 2 Spliting the matrix $\boldsymbol{C}$, one matrix of size 4×4 as a group, then the $\boldsymbol{C} = \{\widehat{\boldsymbol{c}}_1, \widetilde{\boldsymbol{c}}_2, \widehat{\boldsymbol{c}}_3, \widetilde{\boldsymbol{c}}_4, \cdots\}$ in encryption algorithm *Step* 5 can be obtained. Starting from the last group,

$$\begin{cases} \widehat{\boldsymbol{c}}_s \to \overline{\boldsymbol{c}}_s, \boldsymbol{c}_s = \overline{\boldsymbol{c}}_s \oplus \widetilde{\boldsymbol{c}}_{s-1}, & \mathrm{mod}(s, 2) \neq 0, s \neq 1, \\ \widetilde{\boldsymbol{c}}_s \to \overline{\boldsymbol{c}}_s, \boldsymbol{c}_s = \overline{\boldsymbol{c}}_s \oplus \widehat{\boldsymbol{c}}_{s-1}, & \mathrm{mod}(s, 2) = 0, \\ \widehat{\boldsymbol{c}}_1 \to \overline{\boldsymbol{c}}_1, \boldsymbol{c}_1 = \overline{\boldsymbol{c}}_1 \oplus \boldsymbol{c}_S, & s = 1, \end{cases}$$

Algorithm 1: Image preprocessing process

```
Input: $Image$
[$M, N$] ←size($Image$); $sqshu$ ← sqrt(16); $hang$ ← $Image$;
for $i \leftarrow 1 : M$ do
    for $j \leftarrow 2 : N$ do
        $hang(i, j) \leftarrow$ bitxor($hang(i, j), hang(i, j-1)$);
    end
end
$hang1 \leftarrow hang$(:,end);
for $i$ ← 2: length(hang1) do
    $hang1(i) \leftarrow$ bitxor($hang1(i)$, $hang1(i-1)$);
end
$rmin \leftarrow$ 25+mod($hang2$,5); $r \leftarrow rmin$; $rmax \leftarrow rmin$+30; $rh \leftarrow 3$; $hang2 \leftarrow hang1$(end);
    $fig_{\leftarrow}Image$(:); $le \leftarrow$ length($fig\_$); $yu \leftarrow$ mod($le$, $shu$);
if $yu \neq 0$ then
    $yu \leftarrow shu - yu$;
    $fig\_(le + 1 : le + yu) \leftarrow 0$;
end
```

where the rule of $\widehat{\boldsymbol{c}}_s \to \overline{\boldsymbol{c}}_s$ is: the column i moves one position down and the row i moves one position to the right, where $j = 4, 3, 2, 1$; the rule of $\widetilde{\boldsymbol{c}}_s \to \overline{\boldsymbol{c}}_s$ is: the column j moves one position up and the row j moves one position to the left, where $j = 4, 3, 2, 1$. Then we can get $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_S\}$ in encryption algorithm *Step* 4.

Step 3 For the $\boldsymbol{C}$ in *Step* 2, each $\boldsymbol{c}_s$ is arranged into a 1-dimensional sequence c_s to obtain $C = \{c_1, c_2, \cdots, c_S\}$ in encryption algorithm *Step* 3, then

$$\begin{cases} g_s^{'} = c_s/255, \\ g_s = g_s^{'}(g_{s\max}^{'} - g_{s\min}^{'}) + g_{s\min}^{'}, \end{cases}$$

the $g_s \in \{g_1, g_2, \cdots, g_S\}$ in encryption algorithm *Step* 3 can be obtained, start with the last group g_S, each group as the initial condition of the model to decrypt, each group has a corresponding r value during decryption. the r of last group S is r_{end}, the former group is $r_{\text{end}} - 1$, in this order, when $r = r_{\min}$, r starts again from $r_{\max}$ and so on. The $B^{'} = \left\{b_1^{'}, b_2^{'}, \cdots, b_S^{'}\right\}$ in encryption algorithm *Step* 3 can be obtained.

Step 4 The $B^{'}$ in *Step* 3 is denormalized, corresponding to the three cases in encryption *Step* 3:

- If the values are all 0.5 in $b_s^{'}(s = 1, 2, \cdots, S)$, then all values in this group are $b_s = 0$.
- If the values are all equal and not 0.5 in $b_s^{'}(s = 1, 2, \cdots, S)$, then the values in this group are $b_s = b_s^{'} \times 255$.
- If the values are not all equal in $b_s^{'}(s = 1, 2, \cdots, S)$, we use the formula

$$b_s = b_s^{'}(b_{s\max}^{'} - b_{s\min}^{'}) + b_{s\min}^{'}$$

to get the plain text pixel value.

Algorithm 2: Heat flow cryptosystem encryption

```
num_group ← length(fig_)/shu; MiWen ←[];
for i ← 1:num_group do
    index ←(i-1)*shu+1 : $i * shu$; Group ← fig_(index); ave ←mean(Group); for j ←
     1:length(Group) do
        if Group(1:end)=ave then
            a ← 1;
        else
            a ← 2;
        end
    end
    if a=1 then
        if ave=0 then
            for k ← 1:length(Group) do
                Group(k)←0.5;
            end
        else
            gg ← Group(1)/255; for k ← 1:length(Group) do
                Group(k)← gg;
            end
        end
    else
        Group ← double(Group);
        Gmax ← max(Group); Gmin ← min(Group);
        Group ←(Group − Gmin)/(Gmax − Gmin);
    end
    [u22, u2, i33, lel] ← BLICM_jiami(r, Group);
    u22 ← (u22-min(u22))/(max(u22)-min(u22)); u22 ← $u22 * 255$;
    if a=1 then
        if ave=0 then
            uj ← u1-0.5; uj ← round(uj); uj ← uj′;
        else
            uj ← u1*255; uj ← round(uj); uj ← uj′;
        end
    else
        uj ← u1*(Gmax − Gmin)+Gmin; uj ← round(uj); uj ← uj′;
    end
    $r \leftarrow r + rh$;
    if $r \geq rmax$ then
        r ← rmin;
    end
    u22 ← reshape(u22, sqshu, sqshu); u22 ← uint8(u22);
    MiWen ← [MiWen, u22];
end
```

Then the $B = \{b_1, b_2, \cdots, b_S\}$ in encryption algorithm *Step* 2 can be obtained.

Step 5 Merging each 1-dimensional sequence of B in *Step* 4 into a long sequence, we take the first MN elements of the sequence to get the 1-dimensional sequence A in encryption algorithm *Step* 1, we recombine A into a matrix of size $M \times N$ to get the decrypted image.

The diagram of encryption and decryption algorithm is shown in Figure 2.

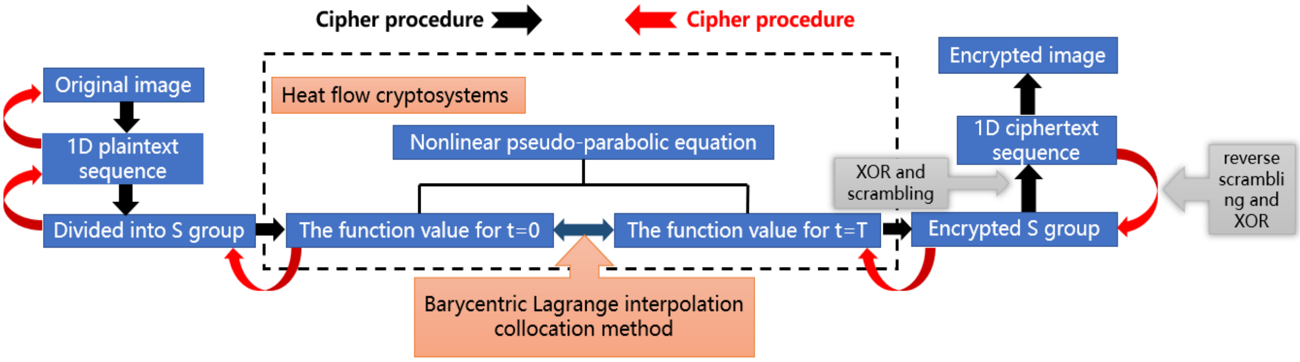


Figure 2. The diagram of encryption algorithm.

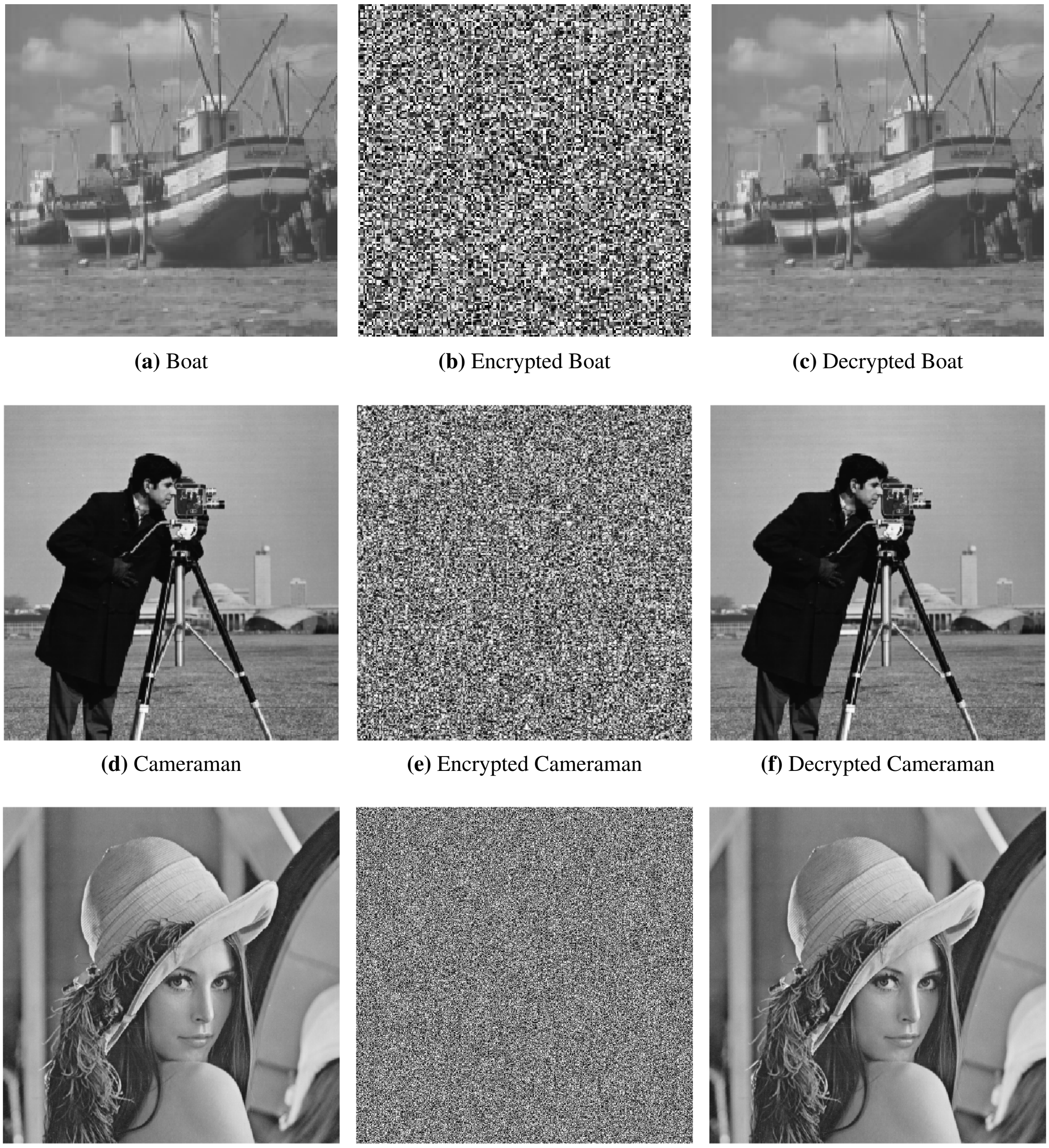
5. Simulation experiment and security analysis

Experimental environment: CPU: i5-11260H, 2.60GH, 64bit operating system, 16G running memory, 512G solid state disk memory. All simulation experiments are realized by the soft Matlab (Version: R2020b).

In this section, (2+1) dimensional encryption and decryption model serve as examples for image encryption experiments, and some gray images and RGB images are used. The key function in (2+1)-dimensional nonlinear pseudo-parabolic equation is selected as

$$\begin{cases} p(x,y) = 1, q(x,y) = \dfrac{10^7}{xy + 0.25}, \\ F(x,y,t,r,u) = 10^9 \cos(0.1 + r(xy + 0.1)t) + 20xyu^2, \end{cases}$$

where $r_{\max} = r_{\min} + 35$, $r_h = 3$, $T = 0.05$, $\Omega = [0,1] \times [0,1]$. We selected three gray images and three RGB images for our experiment.



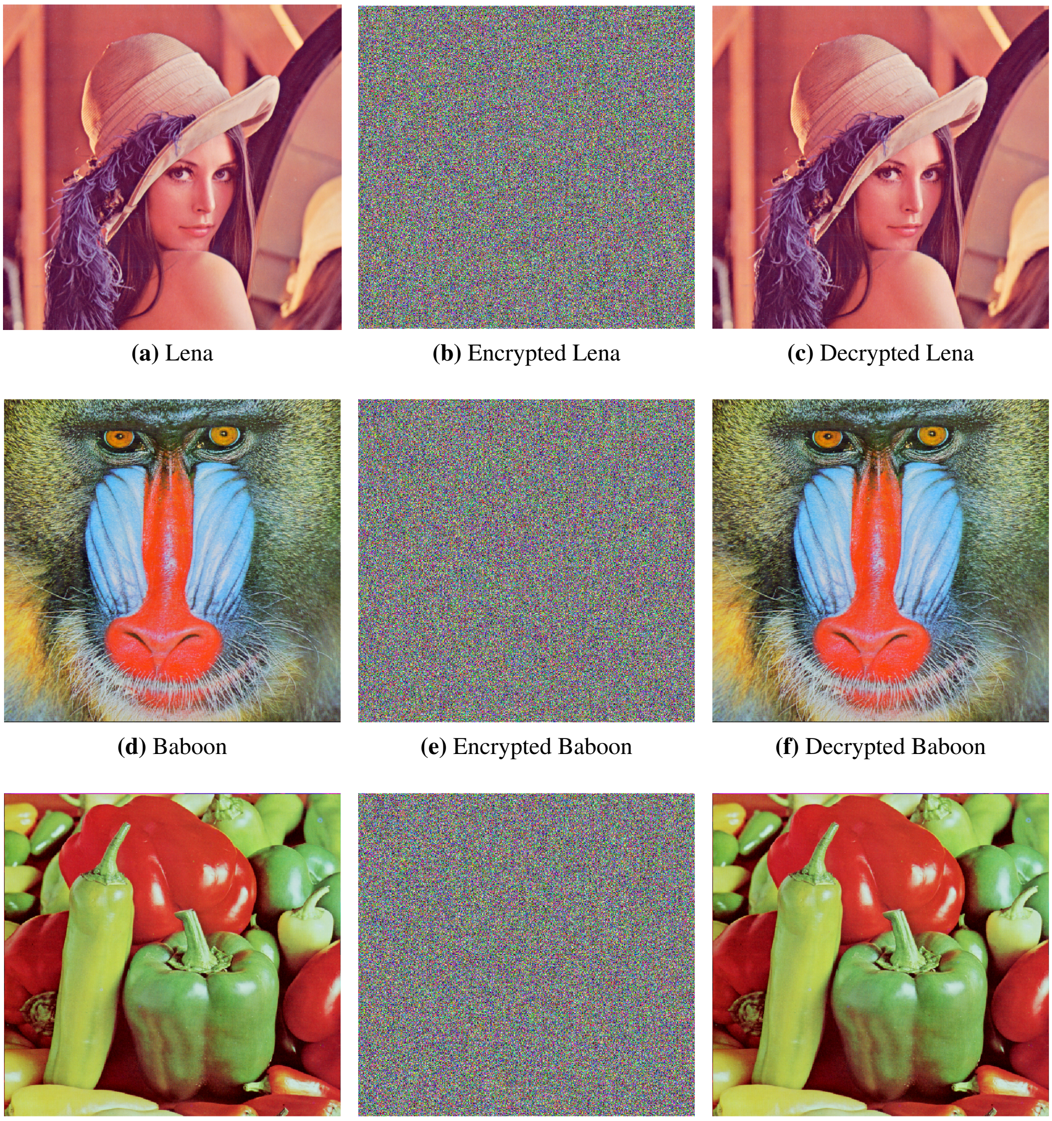
(a) Boat (b) Encrypted Boat (c) Decrypted Boat

(d) Cameraman (e) Encrypted Cameraman (f) Decrypted Cameraman

(g) Lena (h) Encrypted Lena (i) Decrypted Lena

Figure 3. The original image, encrypted image and decrypted image for Boat, Cameraman and Lena.

Figure 3 shows the encrypted and decrypted images of Boat (160 × 160), Cameraman (256 × 256) and Lena (512 × 512) in gray format, respectively, Figure 4 shows the encrypted and decrypted images of Lena (512 × 512), Baboon (512 × 512) and Peppers (512 × 512) in RGB format, respectively, we can see that the encrypted image have no features, the original image can be correctly obtained by the decryption algorithm.



(a) Lena **(b)** Encrypted Lena **(c)** Decrypted Lena

(d) Baboon **(e)** Encrypted Baboon **(f)** Decrypted Baboon

(g) Peppers **(h)** Encrypted Peppers **(i)** Decrypted Peppers

Figure 4. The original image, encrypted image and decrypted image for Lena, Baboon and Peppers.

5.1. Histogram analysis

Histogram can intuitively reflect the distribution of each pixel gray, the pixel distribution of original image is generally uneven. Original image has obvious statistical regularity and is often attacked by statistical analysis attacks. For resist this attack, the histogram of the encrypted image must be uniform to disrupt the regularity of the pixels to prevent the attacker to obtain the useful information. Therefore, the histogram distribution of encrypted image more uniform, the encryption algorithm more better, on the contrary, the encryption algorithm is not safe. Figure 5 shows the gray histograms of the Boat, Cameraman and Lena and their encrypted images and decrypted images. Figures 6–8 shows the histograms of the Lena, Baboon and Peppers in R, G, B components. It can be seen from the histogram that the pixels of encrypted image are completely disrupted, the gray values are evenly distributed.

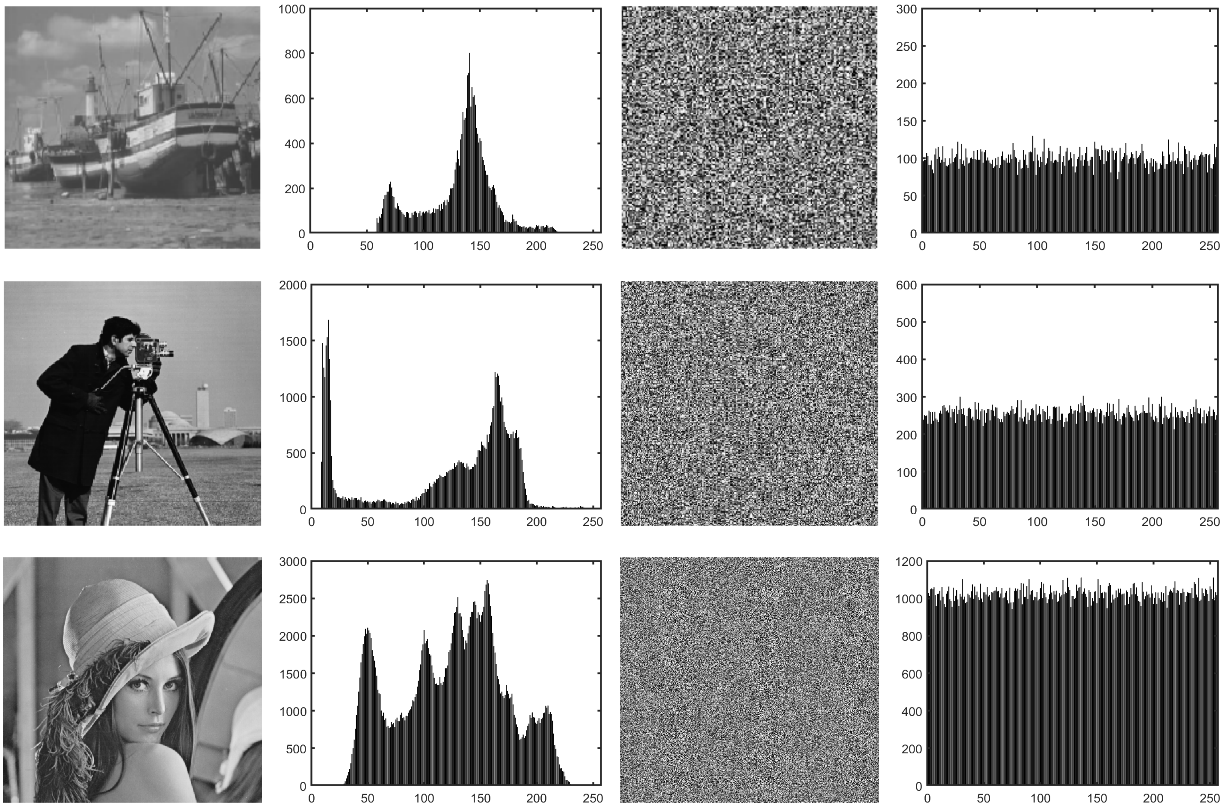
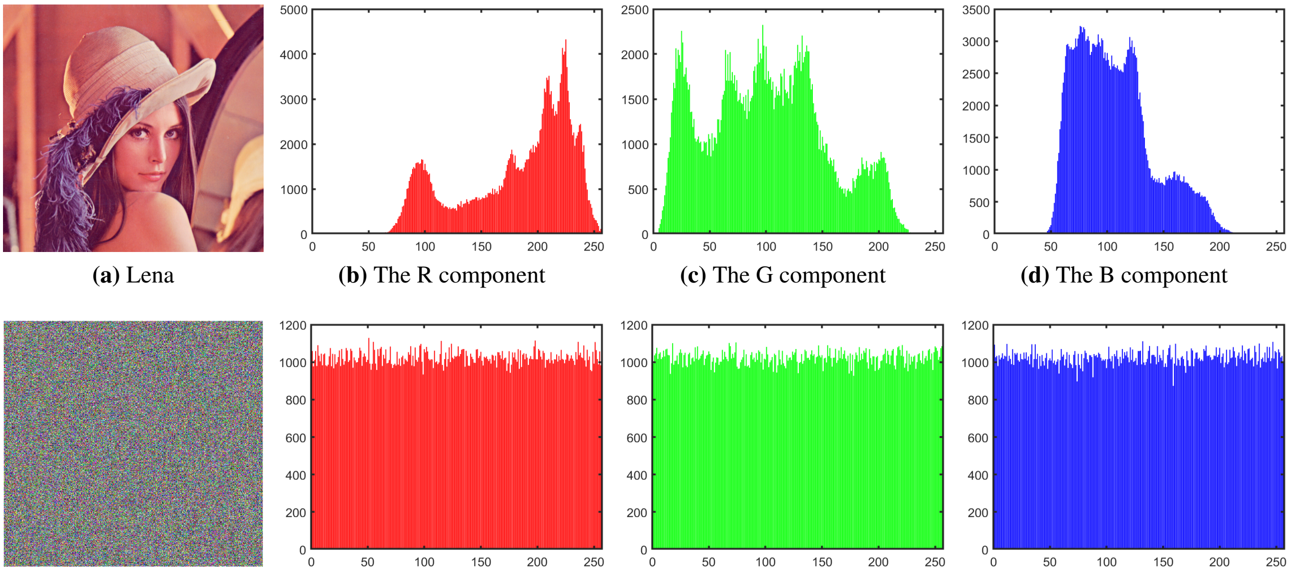


Figure 5. The gray histogram of original image and encrypted image for Boat, Cameraman and Lena.



(a) Lena

(b) The R component

(c) The G component

(d) The B component

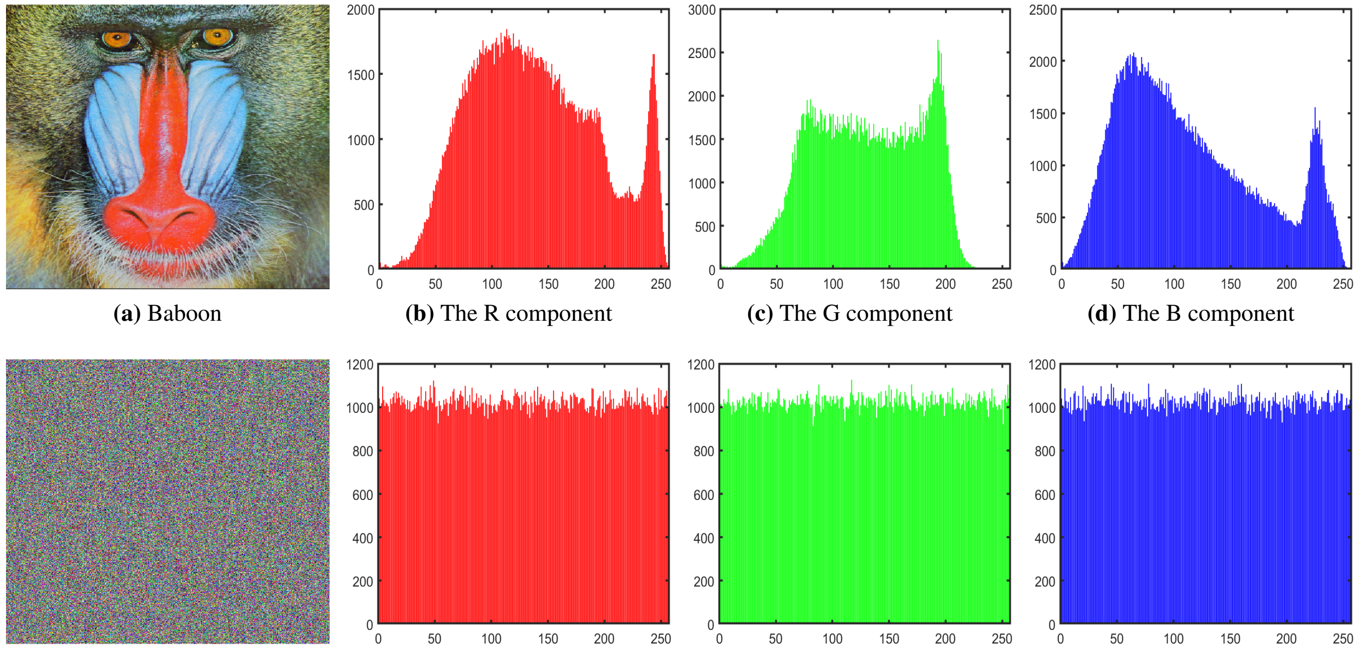
(e) Encrypted Lena

(f) The R component

(g) The G component

(h) The B component

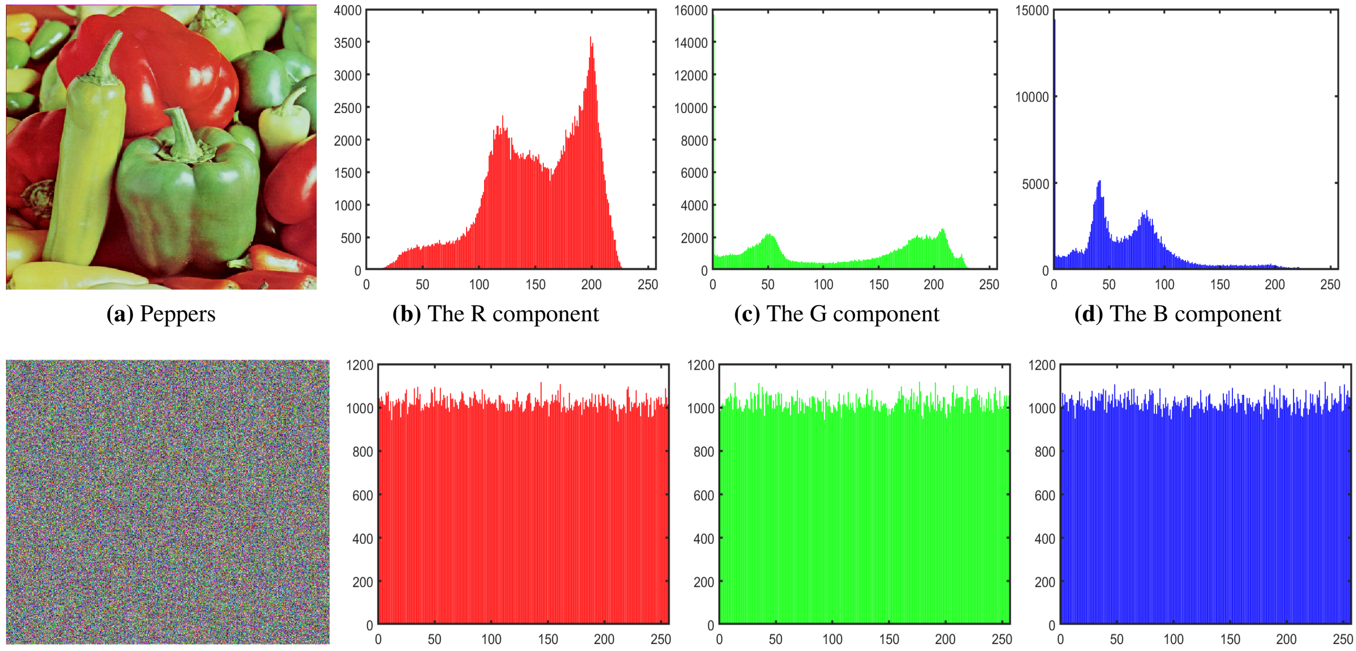
Figure 6. The histogram of original image and encrypted image in RGB component for Lena.



(**a**) Baboon (**b**) The R component (**c**) The G component (**d**) The B component

(**e**) Encrypted Baboon (**f**) The R component (**g**) The G component (**h**) The B component

Figure 7. The histogram of original image and encrypted image in RGB component for Baboon.



(**a**) Peppers (**b**) The R component (**c**) The G component (**d**) The B component

(**e**) Encrypted Peppers (**f**) The R component (**g**) The G component (**h**) The B component

Figure 8. The histogram of original image and encrypted image in RGB component for Peppers.

5.2. Information entropy analysis

Information entropy is an important index to describe information complexity, the confusion and distribution of encrypted image pixels can be quantified by information entropy.

The value of information entropy more larger, the pixel distribution more uniform and the encryption algorithm more security, on the contrary, the value of information entropy more small, the security more worse. The calculation formula of information entropy $H(x)$ as follows

$$H(x) = -\sum_{i=0}^{255} P(x_i)\log_2 P(x_i),$$

where x_i denotes the grey level and $P(x_i)$ is the probability of gray level x_i appearing in this image.

Table 1. Comparison of information entropy between the proposed encryption algorithm and other encryption algorithms for Boat, Cameraman and Lena.

	Original image	The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Boat	6.7252	7.9928	7.9916	7.9890	7.9922	7.9925
Cameraman	7.0097	7.9974	7.9970	7.9955	7.9973	7.9973
Lena	7.4451	7.9992	7.9993	7.9984	7.9993	7.9994

Table 2. Comparison of information entropy between the proposed encryption algorithm and other encryption algorithms for Lena, Baboon and Peppers.

		Original image	The proposed algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Lena	R	7.2531	7.9991	7.9993	7.9990	7.9992	7.9993
	G	7.5940	7.9991	7.9992	7.9990	7.9991	7.9994
	B	6.9684	7.9991	7.9992	7.9967	7.9992	7.9993
Baboon	R	7.7067	7.9992	7.9992	7.9985	7.9991	7.9993
	G	7.4744	7.9993	7.9992	7.9981	7.9994	7.9993
	B	7.7522	7.9992	7.9993	7.9991	7.9993	7.9993
Peppers	R	7.3388	7.9994	7.9993	7.9983	7.9992	7.9992
	G	7.4963	7.9992	7.9993	7.9989	7.9993	7.9993
	B	7.0583	7.9992	7.9993	7.9990	7.9992	7.9993

In a gray image, there are 256 gray levels, the probability of each gray level is 1/256 for ideal encryption, the information entropy can reach the ideal value 8, when the information entropy is closer to the ideal value 8, the encryption effect is more better. Tables 1 and 2 show the comparison of information entropy results between the proposed encryption algorithm and other encryption algorithms. We can see that the information entropy of the proposed encryption algorithm is very close to the ideal value 8, these show the effectiveness of the proposed encryption algorithm.

5.3. Correlation analysis

Correlation generally refers to the degree of correlation between two or more pixels. There is a strong correlation between adjacent pixels of the original image. Encryption algorithms must to disrupt these correlated pixels to achieve the desired encryption effect. The adjacent pixel correlation of encrypted image more weak, the encryption effect more better, on the contrary, if the correlation more strong, the encryption effect more bad. The calculation formula of the correlation coefficient R_{xy} is

$$\begin{cases} R_{xy} = \dfrac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \\ D(x) = \dfrac{1}{MN}\sum_{i=0}^{MN}(x_i - E(x))^2, E(x) = \dfrac{1}{MN}\sum_{i=0}^{MN} x_i, \\ \operatorname{cov}(x,y) = \dfrac{1}{MN}(x_i - E(x))(y_i - E(y)), \end{cases}$$

where x and y are the two adjacent pixels, MN is the sum of pixels in the image.

Table 3. Correlation coefficients between adjacent pixels of Boat and Cameraman in three directions.

		Horizontal	Vertical	Diagonal
Boat	Original image	9.1129e-01	9.1562e-01	8.4806e-01
	Encrypted image	4.7347e-04	4.7251e-04	–7.2513e-03
Cameraman	Original image	9.3276e-01	9.3219e-01	9.0618e-01
	Encrypted image	5.0545e-04	5.0664e-04	2.2202e-03

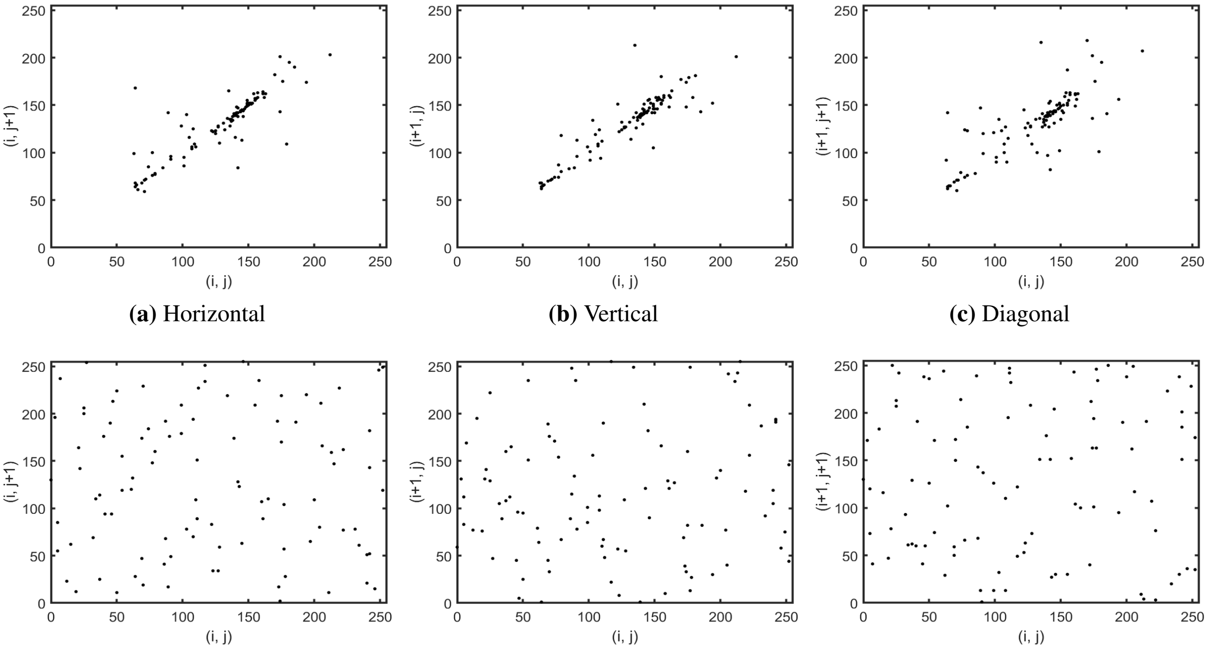
Table 4. Comparison of correlation coefficients of encrypted images with different encryption algorithms for Boat and Cameraman.

		The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Boat	Horizontal	4.7347e-04	–1.0045e-02	9.1929e-03	–4.6696e-03	–9.1234e-03
	Vertical	4.7251e-04	–1.0070e-02	9.1619e-03	–4.6957e-03	–9.1722e-03
	Diagonal	–7.2513e-03	–1.7840e-03	–1.1465e-02	9.3678e-03	1.3823e-02
Cameraman	Horizontal	5.0545e-04	–1.2514e-03	–3.0978e-03	8.1377e-03	7.1051e-03
	Vertical	5.0664e-04	–1.2529e-03	–3.0955e-03	8.1413e-03	7.0914e-03
	Diagonal	2.2202e-03	1.1711e-02	–1.4994e-02	–4.9900e-03	–9.2223e-04

Table 5. Comparison of the average of correlation coefficients in three directions of encrypted images for Lena in gray scale and RGB format.

		The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Gray Lena		1.6165e-03	1.8795e-03	4.2927e-03	2.0797e-03	3.7695e-03
	R	1.0036e-03	2.2689e-03	1.9674e-03	1.2921e-03	3.6633e-03
RGB Lena	G	1.8628e-03	2.5243e-03	3.4400e-03	2.0831e-03	2.4585e-03
	B	3.7065e-04	1.7381e-03	2.0731e-03	4.7438e-03	1.3239e-03

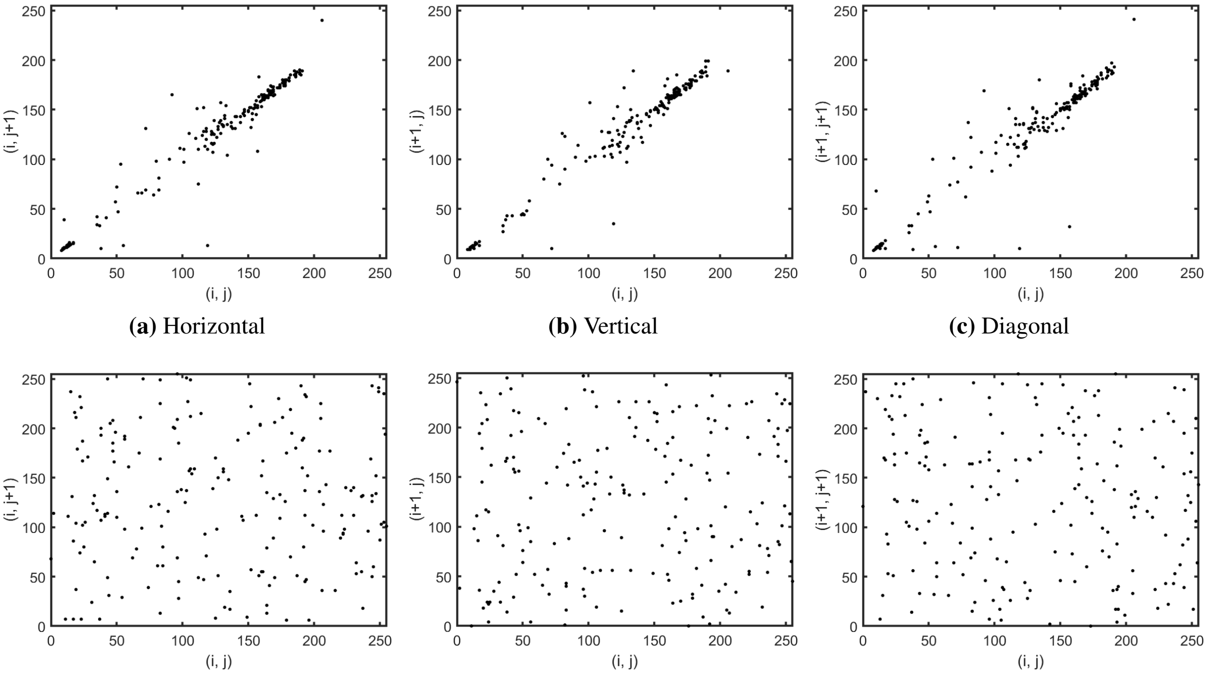
Table 3 shows the correlation coefficients obtained by randomly selecting pixels in the horizontal, vertical and diagonal directions of the original image and encrypted image, where Boat select 100 pixels and Cameraman select 200 pixels. We can see that the correlation coefficients of the encrypted image in all three directions are very close to 0 which show that the encryption algorithm is good to break the correlation of the original pixels. Table 4 is the comparative results of the correlation between different encryption algorithms. In Table 5, we compare the results of Lena in gray and RGB format with other literatures. Observing the results, we can see that most of the correlation coefficient of proposed encryption algorithm are better than Ref. [2, 4–6].



(a) Horizontal **(b)** Vertical **(c)** Diagonal

(d) Horizontal **(e)** Vertical **(f)** Diagonal

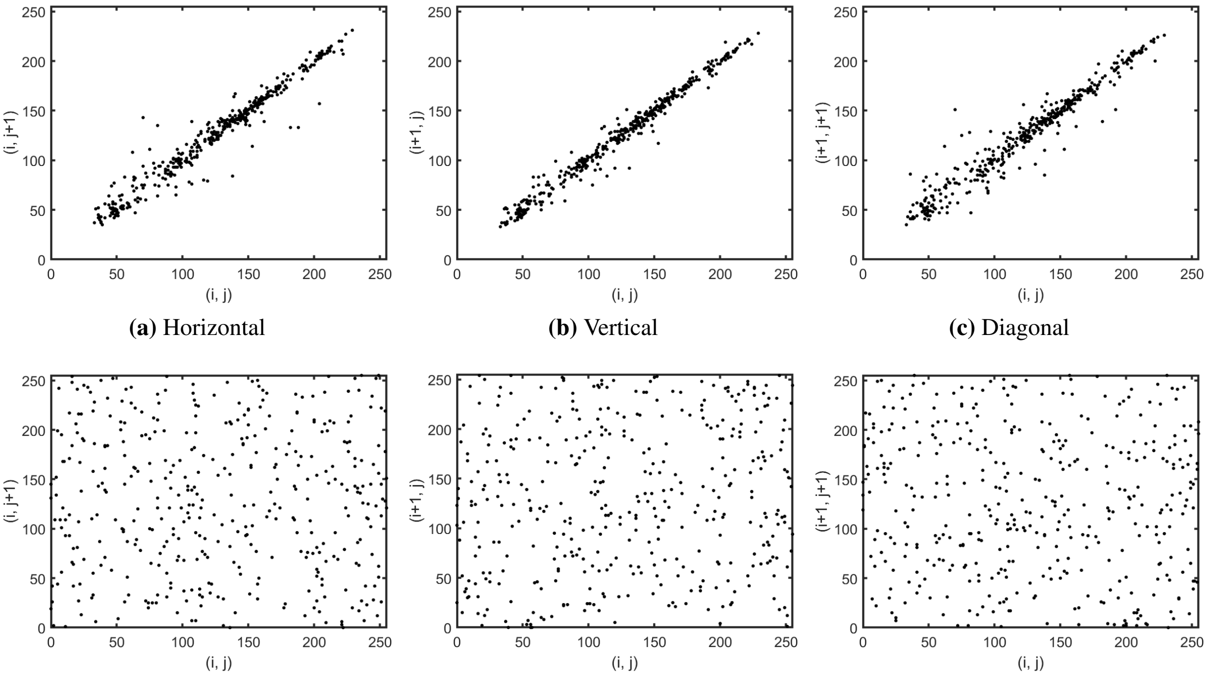
Figure 9. The scatter plot in horizontal, vertical and diagonal directions for Boat and encrypted Boat.



(a) Horizontal **(b)** Vertical **(c)** Diagonal

(d) Horizontal **(e)** Vertical **(f)** Diagonal

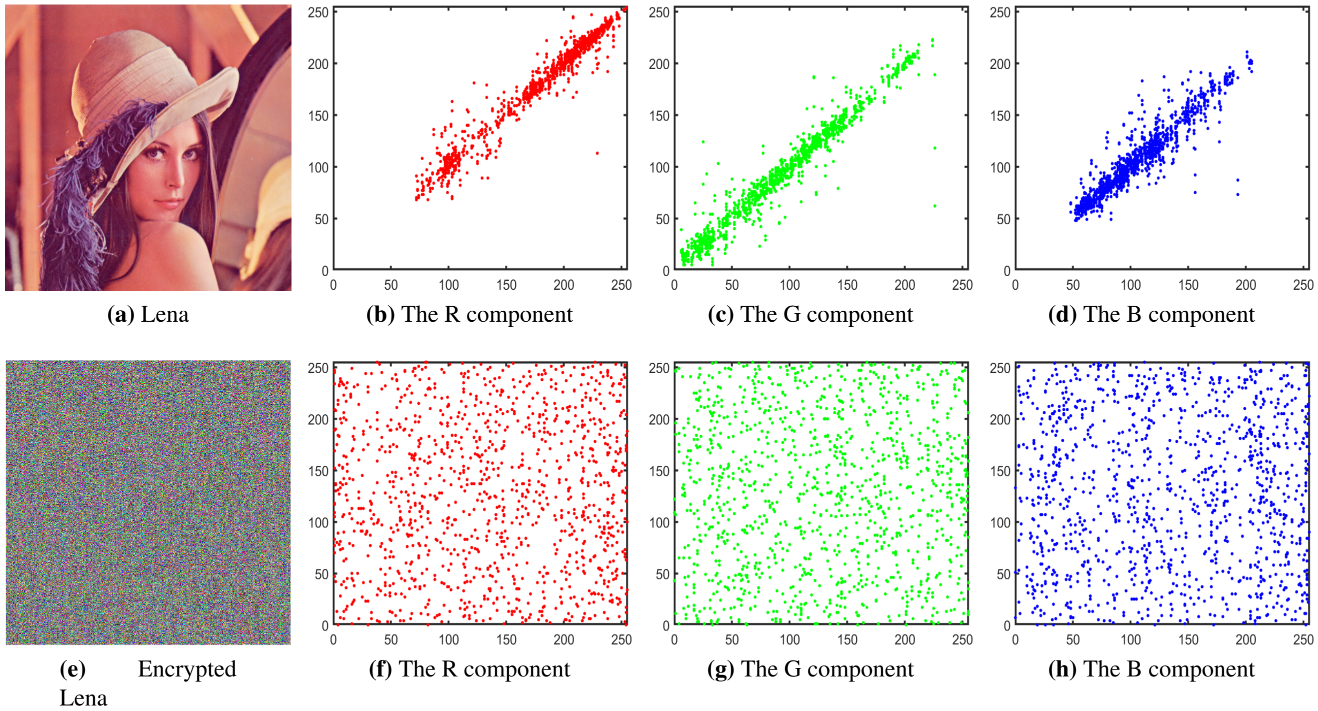
Figure 10. The scatter plot in horizontal, vertical and diagonal directions for Cameraman and encrypted Cameraman.



(a) Horizontal **(b)** Vertical **(c)** Diagonal

(d) Horizontal **(e)** Vertical **(f)** Diagonal

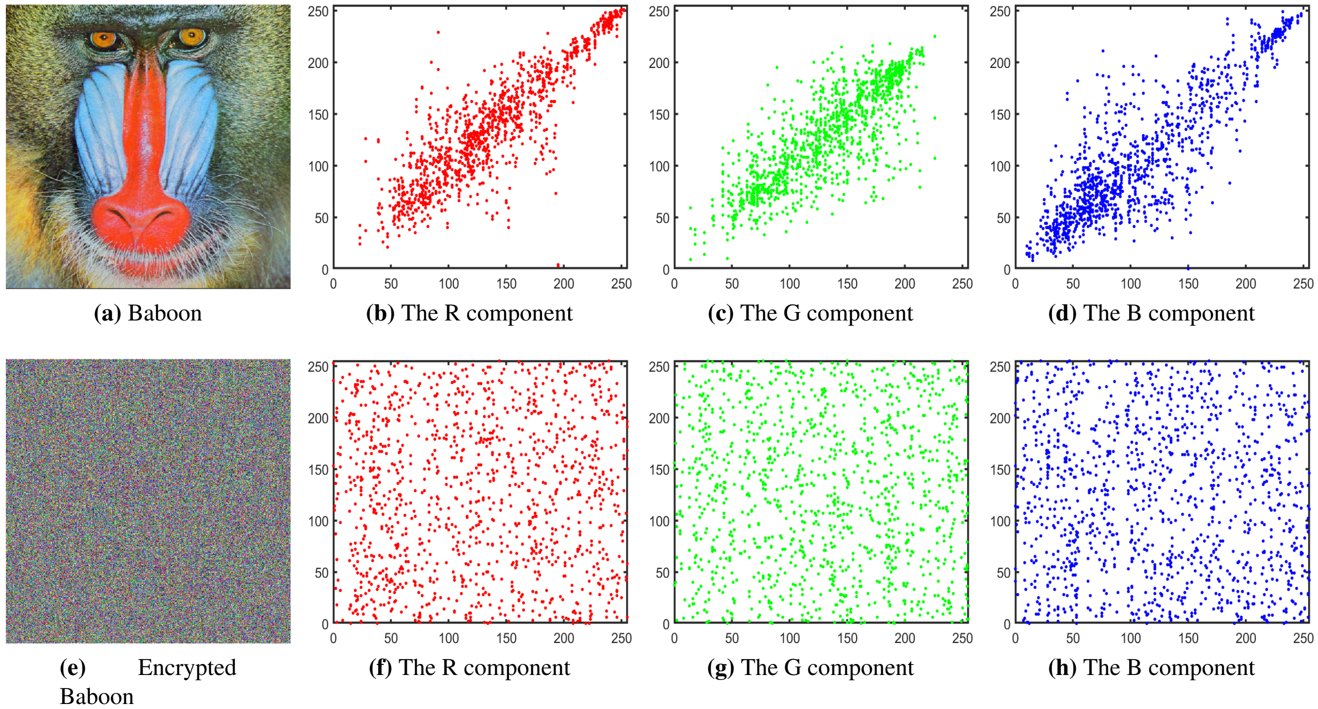
Figure 11. The scatter plot in horizontal, vertical and diagonal directions for Lena and encrypted Lena.



(a) Lena **(b)** The R component **(c)** The G component **(d)** The B component

(e) Encrypted Lena **(f)** The R component **(g)** The G component **(h)** The B component

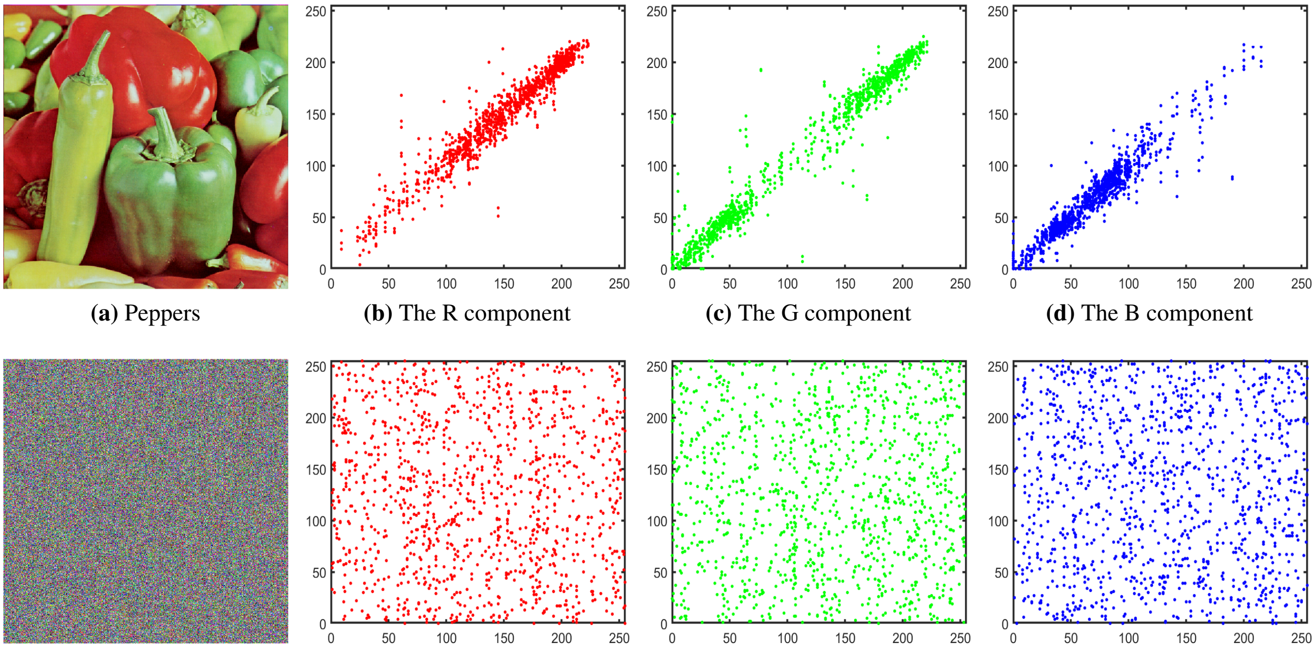
Figure 12. The scatter plot in RGB component for Lena and encrypted Lena.



(a) Baboon **(b)** The R component **(c)** The G component **(d)** The B component

(e) Encrypted Baboon **(f)** The R component **(g)** The G component **(h)** The B component

Figure 13. The scatter plot in RGB component for Baboon and encrypted Baboon.



(a) Peppers **(b)** The R component **(c)** The G component **(d)** The B component

(e) Encrypted Peppers **(f)** The R component **(g)** The G component **(h)** The B component

Figure 14. The scatter plot in RGB component for Peppers and encrypted Peppers.

The correlation distribution of the original image and encrypted image in three directions are shown in Figures 9–11. In these graphs, abscissa denotes the pixel value of position (i, j) in this image, ordinate denotes the pixel values of the horizontal adjacent pixel $(i, j + 1)$, the vertical adjacent pixel $(i + 1, j)$, the diagonal diagonal adjacent pixel $(i + 1, j + 1)$, respectively. In Figures 12–14, we show scatter distribution of the three directions for the RGB image Lena, Baboon and Peppers, in each component, we draw the scatter plots in three directions together. In histogram of original image, we can see that the correlation between adjacent pixels of the original image is very obvious. Observing the scatter distribution in histogram of encrypted image, we can see that the correlation between the pixels of the encrypted image is completely broken and it has not any rule.

5.4. *Sensitivity analysis*

5.4.1. Cipher key sensitivity analysis

Cipher key sensitivity means that when there is a small change in the key, a completely different encryption effect can be produced. An ideal encryption algorithm should be sensitive to cipher key transformation to ensure the security of the encryption algorithm. There are generally two indicators for evaluating key sensitivity namely number of pixels change rate (NPCR) and unified average changing intensity (UACI), respectively. The calculation formulas are

$$\begin{cases} \text{NPCR} = \dfrac{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N} D(i,j)}{MN} \times 100\%, D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j), \\ 0, \text{otherwise}, \end{cases} \\ \text{UACI} = \dfrac{1}{MN} \times \dfrac{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N} |C_1(i,j) - C_2(i,j)|}{255} \times 100\%, \end{cases} \tag{5.1}$$

where C_1 and C_2 are two different encrypted images by use different cipher key. The ideal values of NPCR and UACI are 99.6094% and 33.4635%, respectively. When NPCR and UACI are more closer to the ideal value, the encryption algorithm is more sensitive to the key, the encryption algorithm is more secure. Changing the initial key parameter $r_{\min}$ to $r_{\min}+2$ to encrypt the Boat and Cameraman, the results of NPCR and UACI are shown in Table 6, we can see that the NPCR and UACI of the proposed encryption algorithm are very close to the ideal value, which indicates that the proposed encryption algorithm is sensitive to the key. In addition, the Table 6 gives the comparison of NPCR and UACI obtained by other encryption algorithms. In Table 7, we show the comparison results of Lena in gray and RGB format, for the Lena in RGB format, we use the average value of each component as the final result of NPCR and UACI.

Table 6. Comparison of cipher key sensitivity with different encryption algorithms for Boat and Cameraman.

		The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Boat	NPCR	99.625%	99.629%	99.668%	99.586%	99.590%
	UACI	33.345%	33.281%	34.515%	33.595%	33.472%
Cameraman	NPCR	99.600%	99.646%	99.629%	99.568%	99.631%
	UACI	32.995%	33.531%	33.930%	33.415%	33.341%

Table 7. Comparison of cipher key sensitivity with different encryption algorithms for Lena in gray and RGB format.

		The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Gray Lena	NPCR	99.606%	99.601%	99.582%	99.570%	99.593%
	UACI	33.448%	33.352%	33.870%	33.428%	33.466%
RGB Lena	NPCR(AVG)	99.606%	99.627%	99.582%	99.629%	99.615%
	UACI(AVG)	33.447%	33.420%	33.786%	33.523%	33.480%

5.4.2. Plaintext sensitivity analysis

Differential attack is a means for cracking encrypted image, some attackers use the encryption algorithm to encrypt the original image and the changed original image separately to find the relationship between the original image and the encrypted image. Plaintext sensitivity is mainly to test whether the encryption algorithm can resist this differential attack.

We change one pixel value in the original image, then we use the same key to encrypt the original image and the changed image to get two different ciphertext images. We still use the NPCR and UACI index of the formula (5.1) to measure the difference between two encrypted images. In Tables 8 and 9, we arbitrarily changed the original pixels in the position (1,1), and the results show that the NPCR and UACI is very close to ideal value, which show the reliability of the proposed encryption algorithm. We still use Lena with two formats as an example to show the comparison results of different algorithms, it can be seen that the encryption algorithm in this paper is sensitive to the change of pixels.

Table 8. The NPCR and UACI for Boat.

	124→127	124→131	124→85	124→125
NPCR	99.602%	99.684%	99.629%	99.687%
UACI	33.972%	33.440%	33.591%	33.784%

Table 9. The NPCR and UACI for Cameraman.

	156→50	156→200	156→125	156→152
NPCR	99.657%	99.605%	99.605%	99.577%
UACI	33.428%	33.562%	33.416%	33.324%

Table 10. The NPCR and UACI for Lena in gray and RGB format.

		The proposed encryption algorithm	Ref. [2]	Ref. [4]	Ref. [5]	Ref. [6]
Gray Lena	NPCR	99.615%	16.479%	3.8147e-04%	99.590%	99.612%
	UACI	33.314%	5.5649%	2.9919e-06%	33.421%	20.330%
RGB Lena	NPCR(AVG)	99.621%	20.391%	3.8147e-04%	99.642%	99.612%
	UACI(AVG)	33.577%	6.8034%	2.9919e-06%	33.470%	20.396%

5.4.3. Performance analysis

In Table 11, we compare the results of this paper with the related work in recent years. It can be seen that the effect of the encryption algorithm in this paper can reach the level of the latest research results and has good security.

Table 11. The performance of this paper and other literature on information entropy, correlation coefficient, the NPCR and UACI of key sensitivity for Lena (512 × 512) in gray format.

	H	Horizontal	Vertical	Diagonal	NPCR	UACI
ours	7.9992	–1.5041e-03	–1.5045e-03	5.4874e-04	99.606%	33.448%
Ref. [2]	7.9993	–1.8629e-03	–1.8619e-03	6.1996e-03	99.601%	33.352%
Ref. [4]	7.9984	2.6442e-03	2.6432e-03	–9.3217e-04	99.582%	33.870%
Ref. [5]	7.9993	1.7772e-03	1.7770e-03	3.9794e-03	99.570%	33.428%
Ref. [6]	7.9994	2.3625e-03	2.3609e-03	2.0062e-03	99.593%	33.466%
Ref. [10]	7.9923	1.158e-03	1.98e-04	–2.26e-04	99.594%	33.381%
Ref. [19]	7.9993	–2.1e-03	2.03e-02	8.1e-03	99.809%	33.481%
Ref. [38]	7.7362(AVG)	1.394e-01	3.39e-02	7.55e-02	–	–
Ref. [39]	7.9991	2.2e-03	1.5e-03	2.5e-03	99.61%	33.45%
Ref. [13]	7.9982	1.87e-04	5.92e-04	7.36e-04	–	–
Ref. [9]	7.9993(AVG)	1.2233e-02	7.6667e-03	6.0000e-04	–	–

Table 12. NIST SP800-22 statistical test results.

No.	Test name	P-value	Proportion	Result
1	Frequency	0.739918	29/30	Success
2	Block Frequency	0.602458	30/30	Success
3	Cumulative Sums	0.671779	29/30	Success
4	Runs	0.299251	30/30	Success
5	Longest Run	0.976060	30/30	Success
6	Rank	0.299251	30/30	Success
7	FFT	0.035174	29/30	Success
8	Non-Overlapping Template	0.534146	30/30	Success
9	Overlapping Template	0.804337	30/30	Success
10	Universal	0.949602	30/30	Success
11	Approximate Entropy	0.213309	29/30	Success
12	Random Excursions	0.162606	17/17	Success
13	Random Excursions Variant	0.162606	16/17	Success
14	Serial	0.534146	30/30	Success
15	Linear Complexity	0.299251	30/30	Success

5.4.4. Statistical test NIST SP 800-22

Statistical test NIST SP800-22 is widely used to test the randomness of sequences. We use this standard to test the randomness of the binary sequence size of 30 M generated by the proposed algorithm. The 30 binary sequences are generated and each sequence with a size of 10^6 bits. It can be seen from Table 12 that the binary sequences generated by our algorithm have successfully passed 15 tests. Some information of the test is as follows: the block length M of Block Frequency test is 128, the block length m of Non-Overlapping Template test is 9, the block length m of Overlapping Template test is 9, the block length m of Approximate Entropy test is 10, the block length m of Serial test is 16, the block length M of Linear Complexity test is 500.

6. Results

The heat flow cryptosystem is an encryption technology that is different from traditional encryption methods. In this paper, the cryptosystem is composed of a nonlinear pseudo-parabolic equation and the barycentric Lagrange interpolation collocation method for solving the equation is proposed. An image encryption algorithm based on the heat flow cryptosystem is designed, the image is divided into different groups and each group performs two diffusions and one scrambling. We give several images with gray and RGB format for simulation experiments, including classical Lena, Baboon and so on. Some common image encryption evaluation indexes are proposed to objectively evaluate the image encryption algorithm, such as histogram, information entropy, correlation key sensitivity, etc. Good experimental results are obtained to support our experiment. The experimental results show that the proposed encryption algorithm performs well in most of the indicators and the encryption algorithm is sensitive to the change of key and plaintext.

Heat flow cryptosystem can also be applied to encryption of text, voice, video and other information. In the future work, we will consider the impact of different types of partial differential equations on the encryption effect and how to improve the time efficiency of the model.

Acknowledgments

The work of Jin Li was supported by Natural Science Foundation of Hebei Province (Grant No. A2019209533) and National Natural Science Foundation of China (Grant No. 11771398).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. P. Mahajan, A. Sachdeva, A study of encryption algorithms AES, DES and RSA for security, *Glob. J. Comput. Sci. Technol.*, **13** (2013), 32–40. https://computerresearch.org/index.php/computer/article/view/272
2. K. Y. Sha, Z. X. Wang, Research on image encryption algorithms based on chaos, (in Chinese), *Audio engineering*, **43** (2019), 64–67. https://doi.org/10.16311/j.audioe.2019.01.018
3. W. W. Yuan, C. C. Zhang, Research on chaotic encryption algorithm of binary image, (in Chinese), *Techniques of Automation and Applications*, **41** (2022), 87–90.
4. S. Y. Jiang, G. Y. Wang, P. P. Jin, A new image encryption algorithm based on improved Henon

mapping, (in Chinese), *Journal of Hangzhou Dianzi University (Natural Sciences)*, **37** (2017), 1–6. https://doi.org/10.13954/j.cnki.hdu.2017.05.001

5. K. Zeng, S. M. Yu, Y. C. Hu, Z. Q. Zhang, Image encryption using 3D Logistic-Sine cascade map, (in Chinese), *Application of Electronic Technique*, **46** (2020), 86–91. https://doi.org/10.16157/j.issn.0258-7998.190966

6. J. Y. Liu, J. K. Ge, J. T. Tang, A Fast chaotic image encryption algorithm based on improved sine map, (in Chinese), *Journal of Chongqing University of Science and Technology (Natural Sciences Edition)*, **41** (2022), 87–90. https://doi.org/10.19406/j.cnki.cqkjxyxbzkb.2020.05.016

7. X. Y. Wang, S. N. Chen, Y. Q. Zhang, A chaotic image encryption algorithm based on random dynamic mixing, *Opt. Laser Technol.*, **138** (2021), 106837. https://doi.org/10.1016/j.optlastec.2020.106837

8. S. C. Wang, C. H. Wang, C. Xu, An image encryption algorithm based on a hidden attractor chaos system and the Knuth-Durstenfeld algorithm, *Opt Lasers Eng*, **128** (2020), 105995. https://doi.org/10.1016/j.optlaseng.2019.105995

9. Z. G. Xiong, Y. Wu, C. H. Ye, X. M. Zhang, F. Xu, Color image chaos encryption algorithm combining CRC and nine palace map, *Multimed. Tools. Appl.*, **78** (2019), 31035–31055. https://doi.org/10.1007/s11042-018-7081-3

10. M. M. Guan, X. L. Yang, W. S. Hu, Chaotic image encryption algorithm using frequency-domain DNA encoding, *IET Image Process*, **13** (2019), 1535–1539. https://doi.org/10.1049/iet-ipr.2019.0051

11. Eduardo Rodríguez-Orozco, Enrique Efren García-Guerrero , Everardo Inzunza-Gonzalez, O. R. López-Bonilla, A. Flores-Vergara, J. R. Cárdenas-Valdez, et al., FPGA-based chaotic cryptosystem by using voice recognition as access key, *Electronics*, **7** (2018), 414. https://doi.org/10.3390/electronics7120414

12. X. Wang, X. Chen, An image encryption algorithm based on dynamic row scrambling and Zigzag transformation, *Chaos Solitons Fractals*, **147** (2021), 10962. https://doi.org/10.1016/j.chaos.2021.110962

13. M. L. Sahari, I. Boukemara, A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption, *Nonlinear Dyn.*, **94** (2018), 723–744. https://doi.org/10.1007/s11071-018-4390-z

14. S. Zhou, X. Y. Wang, M. X. Wang, Y. Q. Zhang, Simple colour image cryptosystem with very high level of security, *Chaos Solitons Fractals*, **141** (2020), 110225. https://doi.org/10.1016/j.chaos.2020.110225

15. M. Nazari, M. Mehrabian, A novel chaotic IWT-LSB blind watermarking approach with flexible capacity for secure transmission of authenticated medical images, *Multimed. Tools. Appl.*, **80** (2021), 10615–10655. https://doi.org/10.1007/s11042-020-10032-2

16. P. El-kafrawy, M. Aboghazalah, A. M. Ahmed, Hanaa Torkey, Ayman El-Sayed, An efficient encryption and compression of sensed IoT medical images using auto-encoder, *Comput Model Eng Sci* , **134** (2023), 909–926. https://doi.org/10.32604/cmes.2022.021713

17. D. A. Trujillo-Toledo, O. R. López-Bonilla, E. E. García-Guerrero, J. J. Esqueda-Elizondo, J. R. Cárdenas-Valdez, U. J. Tamayo-Pérez, et al., Real-time medical image encryption for H-IoT applications using improved sequences from chaotic maps, *Integration*, **90** (2023), 131–145. https://doi.org/10.1016/j.vlsi.2023.01.008

18. P. Sarosh, S. A. Parah, G. M. Bhat. An efficient image encryption scheme for healthcare applications, *Multimedia Tools and Applications*, **81** (2022), 7253–7270. https://doi.org/10.1007/s11042-021-11812-0

19. E. E. García-Guerrero, E. Inzunza-González, O. R. López-Bonilla, J. R. Cárdenas-Valdez, Close E. Tlelo-Cuautle, Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels, *Chaos Soliton Fract*, **133** (2020), 109646. https://doi.org/10.1016/j.chaos.2020.109646

20. D. A. Trujillo-Toledo, O. R. López-Bonilla, E. E. García-Guerrero, E. Tlelo-Cuautle, D. López-Mancilla, O. Guillén-Fernández, et al., Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps, *Chaos Solitons Fractals* , **153** (2021), 111506. https://doi.org/10.1016/j.chaos.2021.111506

21. A. Kifouche, M. S. Azzaz, R. Hamouche, R. Kocik, Design and implementation of a new lightweight chaos-based cryptosystem to secure IoT communications, *Int. J. Inf. Secur.*, **21** (2022), 1247–1262. https://doi.org/10.1007/s10207-022-00609-3

22. Z. X. Jia, Y. P. Liu, Novel image encryption algorithm based on self-adaptive diffusion and combined global scrambling, (in Chinese), *Journal of East China Normal University (Natural Science)*, **6** (2019), 61–72. https://doi.org/10.3969/j.issn.1000-5641.2019.06.007

23. H. Ren, S. Z. Niu, R. Y. Ren, Z. Yue, Research on meaningful image encryption algorithm based on 2-dimensional compressive sensing, (in Chinese), *Journal on Communications*, **43** (2022), 45–57. https://doi.org/10.11959/j.issn.1000-436x.2022101

24. X. Y. Wang, L. Feng, H. Y. Zhao, Fast image encryption algorithm based on parallel computing system, *Inf. Sci.*, **486** (2019), 340–358. https://doi.org/10.1016/j.ins.2019.02.049

25. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, A statistical test suite for random and pseudorandom number generators for cryptographic applications, *Booz-allen and hamilton inc mclean va*, (2001).

26. L. Sleem, R. Couturier, TestU01 and Practrand: Tools for a randomness evaluation for famous multimedia ciphers, *Multimed Tools Appl*, **79** (2020), 24075–24088. https://doi.org/10.1007/s11042-020-09108-w

27. W. Marszalek, M. Walczak, J. Sadecki, Two-parameter 0-1 test for chaos and sample entropy bifurcation diagrams for nonlinear oscillating systems, *IEEE Access*, **9** (2021), 22679–22687. https://doi.org/10.1109/ACCESS.2021.3055715

28. G. R. Blakley, W. Rundell, *Cryptosystems Based on an Analog of Heat Flow*, Berlin: Springer Berlin Heidelberg, 1988, 306–329.

29. L. H. Yang, J. Li, The research on the Reproducing Kernel method of Pseudo-parabolic equation in Heat Flow Cryptosystem, (in Chinese), *Natural Science Journal of Harbin Normal University*, **27** (2011), 12–15. https://doi.org/10.3969/j.issn.1000-5617.2010.05.006

30. L. W. Liu, Fourier Pseudo-spectral Method for Some Sobolev Equation and Its Applications in Heat Flow Cryptosystem, (Chinese), Doctoral Thesis of PLA Information Engineering University, Zhengzhou, 2006.

31. N. Li, Y. Guo, W. P. Qin, H. H. Lu, A finite element algorithm used for nonlinear heat flow cryptosystems, (in Chinese), *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, **21** (2001), 43–45. https://doi.org/10.3969/j.issn.1673-5439.2001.03.009

32. G. Q. Gu, Q. Yang, Mixed volume element method of two-dimensional nonlinear pseudo-parabolic equation, (in Chinese), *Science Technology and Engineering*, **11** (2011), 1766–1768. https://doi.org/10.3969/j.issn.1671-1815.2011.08.025

33. C. Z. Gao, H. Tu, H. Y. Song, A Class of heat flow cryptosystems and analysis of results of computer simulations, (in Chinese), *Journal of Information Enqineering University*, **5** (2004), 28–31. https://doi.org/10.3969/j.issn.1671-0673.2004.04.009

34. L. S. Tang, C. S. Jiang, Image encryption algorithm based on heat flow cryptosystem and chaos, (in Chinese), *Computer Engineering and Applications*, **43** (2007), 37–39. https://doi.org/10.3321/j.issn:1002-8331.2007.03.011

35. Z. Q. Wang, S. P. Li, B. T. Tang, Formulations, algorithms and applications on barycentric interpolation in 1D, (in Chinese), *Journal of Shandong Jianzhu University*, **22** (2007), 448–453. https://doi.org/10.3969/j.issn.1673-7644.2007.05.018

36. J. Li, X. N. Su, J. Z. Qu, Linear barycentric rational collocation method for solving telegraph equation, *Math. Methods Appl. Sci.*, **44** (2021), 11720–11737. https://doi.org/10.1002/mma.7548

37. J. Z. Qu, J. Li, X. N. Su, Barycentric Lagrange interpolation collocation method for solving nonlinear, (in Chinese), *Journal of Shandong University (Natural Science)*, (2022). https://kns.cnki.net/kcms/detail/37.1389.N.20221026.1645.002.html

38. H. R. Shakir, An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling, *Multimed. Tools. Appl.*, **78** (2019), 26073–26087. https://doi.org/10.1007/s11042-019-07766-z

39. W. K. Lee, R. C. W. Phan, W. S. Yap, B. M. Goi, SPRING: a novel parallel chaos-based image encryption scheme, *Nonlinear Dyn.*, **92** (2018), 575–593. https://doi.org/10.1007/s11071-018-4076-6

AIMS Press