mathematics

MDPI

Article

Image Encryption Schemes Based on a Class of Uniformly Distributed Chaotic Systems

Hongyan Zang *, Mengdan Tai and Xinyuan Wei

Mathematics and Physics School, University of Science and Technology Beijing, Beijing 100083, China; s20200711@xs.ustb.edu.cn (M.T.); weixy@cgc.org.cn (X.W.)
* Correspondence: a9801255@ustb.edu.cn; Tel.: +86-1312-116-6553

Abstract: This paper proposes a method to construct a one-dimensional discrete chaotic system. First, we define a generalized distance function to control the boundedness of the one-dimensional discrete system. Based on Marotto's theorem, one-dimensional discrete systems are proven to be chaotic in the sense of Li–Yorke, and the corresponding chaos criterion theorem is proposed. The system can be distributed uniformly by adjusting the parameters. In this paper, we propose an image encryption scheme based on a uniformly distributed discrete chaotic system and DNA encoding. DNA encoding and decoding rules are determined by plain text. The experimental results demonstrate that our encryption algorithm has a large key space, high key sensitivity, and fast encryption speed and can resist differential and statistical attacks.

Keywords: chaotic image encryption; uniform distribution; chaotic system; DNA coding

Citation: Zang, H.; Tai, M.; Wei, X. Image Encryption Schemes Based on a Class of Uniformly Distributed Chaotic Systems. *Mathematics* **2022**, *10*, 1027. https://doi.org/10.3390/math10071027

Academic Editor: Lingfeng Liu

Received: 16 February 2022
Accepted: 18 March 2022
Published: 23 March 2022

1. Introduction

Chaos is a special type of complex dynamic behavior displayed in nonlinear systems that commonly exists in nature, such as mathematics, physics, psychology, biology, and other fields. Chaotic systems have many essential properties, such as ergodicity, extreme sensitivity to initial conditions, and good pseudorandom behavior, which makes chaos theory a popular research subject. In recent years, chaotic systems have been rapidly developed and applied in many fields, especially electronic communications and cryptography [1,2]. As one of the most important information carriers, the security of images is very important and has drawn increasing attention from the public and researchers. However, due to a variety of intrinsic characteristics of images, such as a strong correlation of adjacent pixels, data redundancy, and high computational complexity, traditional encryption algorithms are unsuited to encrypt images. Therefore, researchers have proposed many image encryption algorithms [3–5]. Chaos-based image cryptosystems have become one of the most ideal encryption methods [6–8] because of the main features of chaotic systems, such as sensitivity to initial conditions, ergodicity, and highly complex behavior in addition to their mixing properties.

Researchers have been extensively attracted to constructing new chaotic systems in terms of the existing theory [9,10], which involves the discrimination for the existence of chaos in dynamical systems. In 1975, Li and Yorke first defined the term chaos from a mathematical perspective and proposed a criterion for the existence of chaos in one-dimensional discrete dynamical systems [11], which is well known as "period three implies chaos". Under the guidance of Li–Yorke's criterion, Marotto generalized a high-dimensional discrete dynamical system in 1978, which is known as Marotto's theorem [12]. Shi and Chen proposed a new modified version of Marotto's theorem [13] in 2005. Based on Li–Yorke's criterion, a sufficient and necessary condition for the existence of the three periodic points of a quadratic polynomial is obtained by decomposing the real coefficient polynomial in a complex field [14]. A chaos criterion theorem on a cubic discrete system was established

and proven to be chaotic in the sense of Li–Yorke [15]. Nevertheless, it is difficult to construct chaotic systems in terms of this criterion. Thus far, only a small number of related work has been reported. In contrast to Li–Yorke's criterion, Marotto's theorem appears more instrumental in proposing a theoretical direction. Based on Marotto's theorem, Chen and Lai discussed an automatic control problem for discrete-time dynamical systems by adding a control term and then proposed an algorithm to control Lyapunov exponents for discrete-time dynamical systems, which is known as the Chen–Lai algorithm [16]. Moreover, several bounded functions containing modulus, sine, and saw-tooth functions were applied to globally bind the discrete system in the Chen–Lai algorithm, and the control term is a linear function [17]. To eliminate the linear control term in the Chen–Lai algorithm, a chaos criterion theorem for a one-dimensional discrete system was provided, in which a modulus function was used as a bounded function [18]. It is worth considering whether other bounded functions could lead to a similar proposition while eliminating linear control.

Low-dimensional chaotic maps have a simple structure and are easy to implement, but they usually have a small key space. In this paper, the proposed chaotic system overcomes the shortcomings of the traditional low-dimensional chaotic system with a small key space by adjusting parameters; therefore, a uniform distribution can be achieved.

DNA computing technology has attracted more and more attention since Adleman studied it [19]. Cryptography utilizes DNA as an information carrier in image encryption and has shown promising results by taking advantage of excellent DNA properties, such as massive parallelism, large storage, and ultralow power consumption. The authors of [20] proposed a color image encryption scheme based on DNA operations and a spatiotemporal chaotic system. The key stream of image encryption is associated with the key and plaintext image, which improves the ability to resist known plaintext or selected plaintext attack. Liu et al. [21] proposed color image encryption based on dynamic DNA and 4-D memristive hyperchaos. The main feature of the algorithm is that the dynamic DNA mechanism based on hyperchaos is performed on the processes of encoding, confusion, and diffusion, improving the security of the algorithm. Liu et al. [22] combined DNA computing with double-chaos systems composed of Lorenz chaotic mapping with variable parameters and fourth-order Rossler hyperchaotic mapping and proposed an algorithm for color image encryption at the bit level. The double-chaos system compensates for the pseudorandomness of the two types of chaotic mappings, making chaotic sequences more difficult to predict.

In this paper, a generalized distance function is defined as bounded, and it is applied to control the boundedness of a discrete system. In terms of Marotto's theorem, a one-dimensional discrete system is discussed, and the corresponding chaos criterion theorem is set up to determine the existence of chaos in the discrete system. The system can be distributed uniformly by adjusting the parameters. An image encryption scheme is proposed based on this kind of uniformly distributed discrete chaotic system. First, the chaotic sequence is used to scramble and XOR transform the image, and then DNA coding and DNA operation are performed, the rules of which are determined by plain text. The remainder of this paper is organized as follows. Section 2 presents a class of uniformly distributed discrete chaotic systems and analyzes their dynamic behaviors. Image encryption based on DNA coding is proposed in Section 3. Section 4 gives the simulation experiment results and states security analyses. Finally, Section 5 concludes the paper.

2. A Class of Uniformly Distributed Chaotic Systems

2.1. Chen–Lai Algorithm

The Chen–Lai algorithm considers a nonlinear discrete system, not necessarily chaotic, of the form:

$$x_{k+1} = f_k(x_k),\ x_k \in R^n. \tag{1}$$

Then, a control input sequence $\{u_k\}_{k=0}^{\infty}$ is designed to investigate the automatic control of system (1), and a new system is described as:

$$x_{k+1} = f_k(x_k) + u_k,\ x_k \in R^n, \tag{2}$$

where $u_k = B_k x_k$ is discussed for short.

Assume the sequence $\{B_k\}$ is uniformly bounded:

$$\sup_{0 \le k < \infty} \|B_k\| \le M < \infty$$

where M is a positive constant and $\|\cdot\|$ denotes the spectral norm of a finite-dimensional matrix.

Under the only assumption, it is proven that, in practice, the algorithm can provide the required Lyapunov exponents and achieve the expected anti-control of system (2).

The Chen–Lai algorithm, based on the modulus, sine, and saw-tooth function, is further discussed in detail in [17]. Based on the modulus operation, the one-dimensional discrete system in the Chen–Lai algorithm has the form:

$$x_{k+1} = f(x_k) + u_k \ (\mathrm{mod} 1), \tag{3}$$

where $u_k = (N + e^c)x_k$ is the control term and N and c are two constants.

A proposition is given on the one-dimensional discrete system (3) as follows.

Lemma 1. *[17] If $f(0) = 0$, $c > 0$ and $|f'(x)| < 1 \le N$ are satisfied, then the controlled system (3) is chaotic in the sense of Li–Yorke.*

The chaotic systems constructed by proposition 1 appear limited in having a linear control term; then, a one-dimensional system without a linear control term is considered with the form [18]:

$$x_{k+1} = f(x_k) \ \ (\mathrm{mod} 1), \tag{4}$$

where $x_k \in R^1$, $f(x) \in C^1[0,1]$ and $f(0) = 0$.

The chaos criterion theorem for system (4) is also provided.

Lemma 2. *[18] If $|f'(x)| > 1$ and $x \in [0,1)$ are satisfied, then system (4) is chaotic in the sense of Li–Yorke.*

Evidently, Lemma 1 is a special case of Lemma 2. In short, the form of system (3) is generalized to the form of system (4), while the bounded function is a modulus function. Naturally, under the guidance of the Chen–Lai algorithm and the work of proposition 2, a new bounded function can be defined to replace the modulus function and propose the corresponding chaos criterion theorem. This work is described in Section 2.2.

2.2. A One-Dimensional Discrete Chaotic System

This definition begins without describing the modified Marotto's theorem, which is used later.

Lemma 3. *[13] Let $f : R \to R$ be a map with a fixed point $z \in R^n$. Assume that*

(1) f is continuously differentiable in a neighborhood of z and all the eigenvalues of $Df(z)$ have absolute values larger than 1, which implies that there exists a positive constant r and a norm $\|\cdot\|$ in R^n such that f is expanding in $\overline{B}_r(z)$ in $\|\cdot\|$, where $\overline{B}_r(z)$ is the closed ball of radius centered at z in $(R^n, \|\cdot\|)$;

(2) z is a snap-back repeller of f with $f^m(x_0) = z$, $x_0 \ne z$, for some $x_0 \in B_r(z)$ and some positive integer m, where $B_r(z)$ is the open ball of radius centered at z in $(R^n, \|\cdot\|)$. Furthermore, f is continuously differentiable in some neighborhoods of $x_0, x_1, \cdots, x_{m-1}$, and $\det Df(x_j) \ne 0$ for $0 \le j \le m-1$, where $x_j = f(x_{j-1})$ and $0 \le j \le m-1$.

2.2.1. A Generalized Distance Function

First, a distance function is defined.

Definition 1. *Let $x \in R$; then, there exists an integer $N \in R$ such that $x \in [N, N+1]$. A distance function as $f(x) = \min\{x - N, N + 1 - x\}$ is defined. For simplicity, it is denoted as $f(x) \triangleq (x)$.*

Absolutely, the function $f(x) \triangleq (x)$ is an even function. Then, a generalized distance function is further defined by adding a scale parameter into the distance function in Definition 1.

Definition 2. *A generalized distance function is defined as $Dis_\varepsilon(x) = \varepsilon \cdot (x)$, where parameter ε is a positive constant.*

The image of function $Dis_\varepsilon(x)$ is displayed in Figure 1. Note that $Dis_\varepsilon(x) \in [0, \varepsilon/2]$ and it is also an even function.

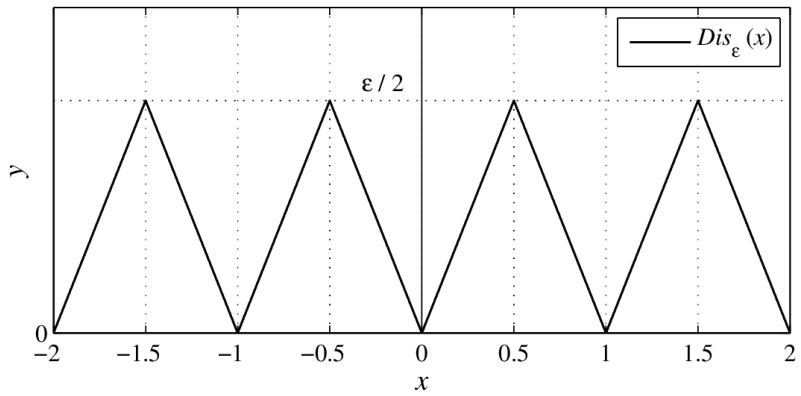


Figure 1. Image of function $Dis_\varepsilon(x)$.

2.2.2. Two Chaos Criterion Theorems

Theorem 1. *Consider a one-dimensional linear discrete system.*

$$x_{k+1} = Dis_\varepsilon(ax_k),\ a \neq 0. \tag{5}$$

If $|a|\varepsilon \geq 2$ is satisfied, then system (5) is a chaotic system in sense of Li–Yorke.

Proof of Theorem 1. If $a < 0$ is satisfied, then $x_{k+1} = Dis_\varepsilon(ax_k) = Dis_\varepsilon(-ax_k)$, which is actually $a > 0$. Therefore, only $a > 0$ is proved for simplicity.□

Denote $g(x) = Dis_\varepsilon(ax)$. Then, while $0 \leq x \leq 1/a$, $g(x)$ can be expressed as:

$$g(x) = \begin{cases} a\varepsilon x, & 0 \leq x < 1/(2a) \\ \varepsilon(1 - ax), & 1/(2a) \leq x \leq 1/a \end{cases}$$

The derivative of map $g(x)$ satisfies $|g'(x)| = |a|\varepsilon > 1$, where $0 \leq x \leq 1/a$ and $x \neq 1/(2a)$. In addition, condition $|a|\varepsilon = a\varepsilon \geq 2$ gives $1/a \leq \varepsilon/2$.

Denote $J_1 = (0, 1/(2a))$, $J_2 = (1/(2a), 1/2a)$.

In interval J_2, $g(x) = \varepsilon(1 - ax) = x$ gives a fixed point:

$$x^* = \frac{\varepsilon}{1 + a\varepsilon}$$

Next, the fixed point x^* is proven as a snap-back repeller.

A sequence $\{x_{-m}|m = 0, 1, 2, \cdots\}$ is defined as:

$$x_0 = \frac{1}{a(1+a\varepsilon)} \in J_1,\ x_{-m} = \frac{\varepsilon - x_{-m+1}}{a\varepsilon} \in J_2,\ m = 1, 2, \cdots$$

Then, $x^* = g(x_0) = g^2(x_{-1}) = \cdots = g^m(x_{-m+1})$.

By the Lagrange mean value theorem, there exists a point $\xi \in (x_{-m+1}, x^*)$ or $\xi \in (x^*, x_{-m+1})$ such that:

$$\left|x_{-m+2} - x^*\right| = \left|g(x_{-m+1}) - g(x^*)\right| = \left|g'(\xi)(x_{-m+1} - x^*)\right| > \left|x_{-m+1} - x^*\right|$$

Hence, let the positive integer m be large enough; then, there exists a constant $r > 0$ such that:

$$x_{-m+1} \in B_r(x^*) \subset J_2,\ x_{-m+k} \notin B_r(x^*),\ k = 2, 3, \cdots, m$$

where $B_r(x^*) = [x^* - r, x^* + r]$ is a closed ball and $g(x)$ is continuously differentiable in $B_r(x^*)$.

In summary, the fixed point x^* satisfies the following conditions:

(a) There exists a positive constant $r > 0$ such that for any point $x \in B_r(x^*) \subset J_2$,

$$\det\{Dg(x)\} = \left|g'(x)\right| = |a|\varepsilon = a\varepsilon > 1,$$

where $Dg(x) = g'(x)$ denotes the Jacobi matrix of $g(x)$.

That is, the eigenvalue of $Dg(x)$ is $\lambda = g'(x)$, and it satisfies $|\lambda| = |g'(x)| = a\varepsilon > 1$.

(b) There exists a point $x_{-m+1} \in B_r(x^*) \subset J_2$ and a positive integer $m \geq 2$ such that $g^m(x_{-m+1}) = x^*$, and

$$|\det\{Dg^m(x_{-m+1})\}| = \left|\prod_{i=1}^{m} \det\{Dg(x_{-i+1})\}\right| = (a\varepsilon)^m \neq 0$$

In summary, x^* is a snap-back repeller of system (5). This completes the proof.

In the proof of Theorem 1, the derivative of map $g(x) = Dis_\varepsilon(ax)$ satisfies $|g'(x)| = |a\varepsilon| > 2$, which shows that the Lyapunov exponent of system (5) is $\lambda = \ln|a\varepsilon| > 0$.

Theorem 2. *Consider a one-dimensional nonlinear discrete system*

$$x_{k+1} = Dis_\varepsilon(f(x_k)), \tag{6}$$

where $f(x) \in C^1[0, \varepsilon/2]$ *and* $f(0) = 0$.

If $|f'(x)| > 1$, $x \in [0, \varepsilon/2]$ *and* $\varepsilon \geq 2$, *then system (6) is chaotic in the sense of Li–Yorke.*

Proof of Theorem 2. Assume $|f'(x)| > 1$ gives $f'(x) > 1$ or $f'(x) < -1$.□

If $f'(x) < -1$ is satisfied, then $x_{k+1} = Dis_\varepsilon(f(x_k)) = Dis_\varepsilon(-f(x_k))$, which is actually $f'(x) > 1$. Therefore, only $f'(x) > 1$ is proved for simplicity.

Denote $g(x) = Dis_\varepsilon(f(x))$, then its derivative satisfies $|g'(x)| = |Dis_\varepsilon{}'(x)f'(x)| = \varepsilon f'(x) > 1$.

By the Lagrange mean value theorem, there exists a point $\xi \in [0, x]$ such that

$$f(x) = f(x) - f(0) = f'(\xi)x > x$$

which gives $f(1/2) > 1/2$ and $f(1) > 1$.

Since $f(1/2) > 1/2 > 0 = f(0)$, there exists a point $t_0 \in (0, 1/2)$ such that $f(t_0) = 1/2$.

Since $f(1) > 1 > 1/2 = f(t_0)$, there exists a point $t_1 \in (t_0, 1)$ such that $f(t_1) = 1$.

The monotonicity of map $f(x)$ ensures the uniqueness of points t_0 and t_1.

Then, $g(t_0) = Dis_\varepsilon(f(t_0)) = \varepsilon/2$ and $g(t_1) = Dis_\varepsilon(f(t_1)) = 0$.

Denote $h(x) = g(x) - x$, and $\varepsilon \geq 2$ gives

$$h(t_0) = g(t_0) - t_0 = \varepsilon/2 - t_0 > 0, h(t_1) = g(t_1) - t_1 = -t_1 < 0$$

which indicates there exists a point $x^* \in (t_0, t_1)$ such that $h(x^*) = 0$. Moreover, $h'(x) = g'(x) - 1 > 0$.

Absolutely, point x^* is a fixed point of map $g(x)$ in interval (t_0, t_1).

Next, the fixed point x^* is proven to be a snap-back repeller.

Since $g(0) = 0 < x^* < \varepsilon/2 = g(t_0)$, there exists a point $x_0 \in (0, t_0)$ such that $g(x_0) = x^*$.

Since $g(t_1) = 0 < x_0 < x^* = g(x^*)$, there exists a point $x_{-1} \in (x^*, t_1)$ such that $g(x_{-1}) = x_0$.

Since $g(x^*) = x^* < x_{-1} < \varepsilon/2 = g(t_0)$, there exists a point $x_{-2} \in (t_0, x^*)$ such that $g(x_{-2}) = x_{-1}$.

Since $g(t_1) = 0 < x_{-2} < x^* = g(x^*)$, there exists a point $x_{-3} \in (x^*, t_1)$ such that $g(x_{-3}) = x_{-1}$.

Therefore, it can be proven that there exists a point $x_{-m+1} \in (t_0, x^*)$ or $x_{-m+1} \in (x^*, t_1)$ such that $g(x_{-m+1}) = x_{-m+2}$.

Then, $x^* = g(x_0) = g^2(x_{-1}) = \cdots = g^m(x_{-m+1})$.

By the Lagrange mean value theorem, there exists a point $\xi \in (x_{-m+1}, x^*)$ or $\xi \in (x^*, x_{-m+1})$ such that

$$\left|x_{-m+2} - x^*\right| = \left|g(x_{-m+1}) - g(x^*)\right| = \left|g'(\xi)(x_{-m+1} - x^*)\right| > \left|x_{-m+1} - x^*\right|.$$

Hence, let the positive integer m be large enough; then, there exists a constant $r > 0$ such that

$$x_{-m+1} \in B_r(x^*) \subset (t_0, t_1), x_{-m+1} \notin B_r(x^*), k = 2, 3, \cdots, m$$

where $B_r(x^*) = [x^* - r, x^* + r]$ is a closed ball and $g(x)$ is continuously differentiable in $B_r(x^*)$.

In summary, the fixed point x^* satisfies the following conditions:

(a) There exists a positive constant $r > 0$ such that for any point

$$x \in B_r(x^*) \subset (t_0, t_1), \det\{Dg(x)\} = \left|g'(x)\right| = \varepsilon f'(x) > 1,$$

where $Dg(x) = g'(x)$ denotes the Jacobi matrix of $g(x)$.

That is, the eigenvalue of $Dg(x)$ is $\lambda = g'(x)$, and it satisfies $|\lambda| = |g'(x)| = \varepsilon f'(x) > 1$.

(b) There exists a point $x_{-m+1} \in B_r(x^*) \subset (t_0, t_1)$ and a positive integer $m \geq 2$ such that $g^m(x_{-m+1}) = x^*$, and

$$\left|\det\{Dg^m(x_{-m+1})\}\right| = \left|\prod_{i=1}^{m} \det\{Dg(x_{-i+1})\}\right| = (\varepsilon f'(x))^m \neq 0.$$

In summary, x^* is a snap-back repeller of system (6). This completes the proof.

Theorem 2 cannot contain Theorem 1 while setting $f(x) = ax$ in Theorem 2.

2.2.3. Three Specific Propositions

To explain the application of Theorem 2, three propositions, based on Theorem 2, are proposed by designing the form of $f(x)$ in system (6).

Proposition 1. *Consider a one-dimensional discrete system*

$$x_{k+1} = Dis_\varepsilon({x_k}^2 + ax_k) = Dis_\varepsilon(f(x_k)), \tag{7}$$

where $f(x) = x^2 + ax$ and $\varepsilon \geq 2$. If $a > 1$ or $a < -\varepsilon - 1$, then system (7) is chaotic.

Note that $|f'(x)| = |2x + a| > 1$ and $x \in [0, \varepsilon/2]$ give $a > 1$ or $a < -\varepsilon - 1$.

Proposition 2. *Consider a one-dimensional discrete system*

$$x_{k+1} = Dis_\varepsilon(a_n x_k^n + a_{n-1} x_k^{n-1} + \cdots + a_1 x_k) = Dis_\varepsilon(f(x_k)), \tag{8}$$

where $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x$ *and* $\varepsilon \geq 2$. *If* $|f'(x)| > 1$, $x \in [0, \varepsilon/2]$, *then system (8) is chaotic.*

Especially, if set $a_i > 0$, $i = 2, 3, \cdots, n$ *and* $a_1 > 1$, *then* $f'(x) = a_n x^{n-1} + \cdots + a_2 x + a_1 > 1$, $x \in [0, \varepsilon/2]$. *That is, system (8) is chaotic.*

Proposition 3. *Consider a one-dimensional discrete system*

$$x_{k+1} = Dis_\varepsilon\left(\int_0^{x_k} g(t)dt + a x_k\right) = Dis_\varepsilon(f(x_k)), \tag{9}$$

where $f(x) = \int_0^x g(t)dt + ax$, $g(x) \in C^1[0, \varepsilon/2]$ *and* $\varepsilon \geq 2$. *If* $g(x) > 0$, $x \in [0, \varepsilon/2]$ *and* $a > 1$, *then system (9) is chaotic. Note that* $f'(x) = g(x) + a > 0 + 1 = 1$, $x \in [0, \varepsilon/2]$ *and* $f(0) = 0$.

2.3. Dynamical Properties Analysis

In this section, the dynamic properties of chaotic systems in Theorems 1 and 2 will be analyzed by means of numerical simulations, such as bifurcation diagrams, and Lyapunov exponent spectra. Bifurcation diagrams describe the process in which states of nonlinear systems change when one parameter changes. An examination of Lyapunov exponents and bifurcation diagram together proved the existence of the chaotic behavior feature.

2.3.1. Bifurcation Diagrams and Lyapunov Exponent Spectra

In Theorem 1, let $\varepsilon = 2$; then, if $|a| \geq 1$, system (5) is chaotic in the sense of Li–Yorke. Let $a = 1$; then, if $\varepsilon \geq 2$, system (5) is chaotic in the sense of Li–Yorke.

Figure 2a,b show the bifurcation diagram and Lyapunov exponent spectrum of parameterin system (5) respectively, where. Figure 2c,d show the bifurcation diagram and Lyapunov exponent spectrum of parameterin system (5) respectively, where. As shown in Figure 2, system (5) displays its chaotic characteristics as theorem 1 expects.

In Proposition 1, let $\varepsilon = 2$; then, if or, system (7) is chaotic. The bifurcation diagram and Lyapunov exponent spectrum of the parameter in system (7) are shown in Figure 3a,b, respectively. Figure 3 shows that system (7) displays chaotic characteristics as expected by Proposition 1.

In Proposition 3, set $g(x) = \sin(x) + 1$; then, if $a > 1$, system (9) is chaotic. The bifurcation diagram and Lyapunov exponent spectrum of parameter a in system (9) are shown in Figure 4a,b, respectively. Figure 4 shows that system (9) displays chaotic characteristics expected by Proposition 3.

2.3.2. Correlation Analysis

In this subsection, we set $a = 1$, $\varepsilon = 3$, $x_0 = 0.2759$ in system (5).

We set $a = 2$, $\varepsilon = 2$, $x_0 = 0.2759$ in system (7).

The evolution of the state variable $k - x(k)$ in systems (5) and (7) for the first 3000 iterations is shown in Figure 5a,b, respectively. The dynamic behaviors of chaotic systems (5) and (7) all demonstrate chaotic attractors.

Autocorrelation and cross-correlation are two main methods to measure the pseudorandomness of chaotic systems. For a truly random series such as white noise, the autocorrelation and cross-correlation are the δ function and zero, respectively.

The autocorrelation coefficient at lag k of a series $\{x(n)\}$ of length N is normally given as:

$$autocorr(k) = \frac{\sum_{i=1}^{N} (x(i) - \overline{x})(x(i+k) - \overline{x})}{\sum_{i=1}^{N} (x(i) - \overline{x})^2},$$

where $\overline{x}$ is the mean of the series $\{x(n)\}$.

The cross-correlation of two series $\{x(n)\}$ and $\{y(n)\}$ of length N at lag k is defined as:

$$crosscorr(k) = \frac{\sum_{i=1}^{N}(x(i)-\overline{x})(y(i-k)-\overline{y})}{\sqrt{\sum_{i=1}^{N}(x(i)-\overline{x})^2}\sqrt{\sum_{i=1}^{N}(y(i)-\overline{y})^2}}.$$

For system (5), the autocorrelation function of the chaotic sequence generated with the initial parameters $a = 1$, $\varepsilon = 3$, and $x_0 = 0.2759$ is shown in Figure 6a, and its cross-correlation with another chaotic system generated with $a = 1$, $\varepsilon = 3$, and $x_0 = 0.3257$ is shown in Figure 6b.

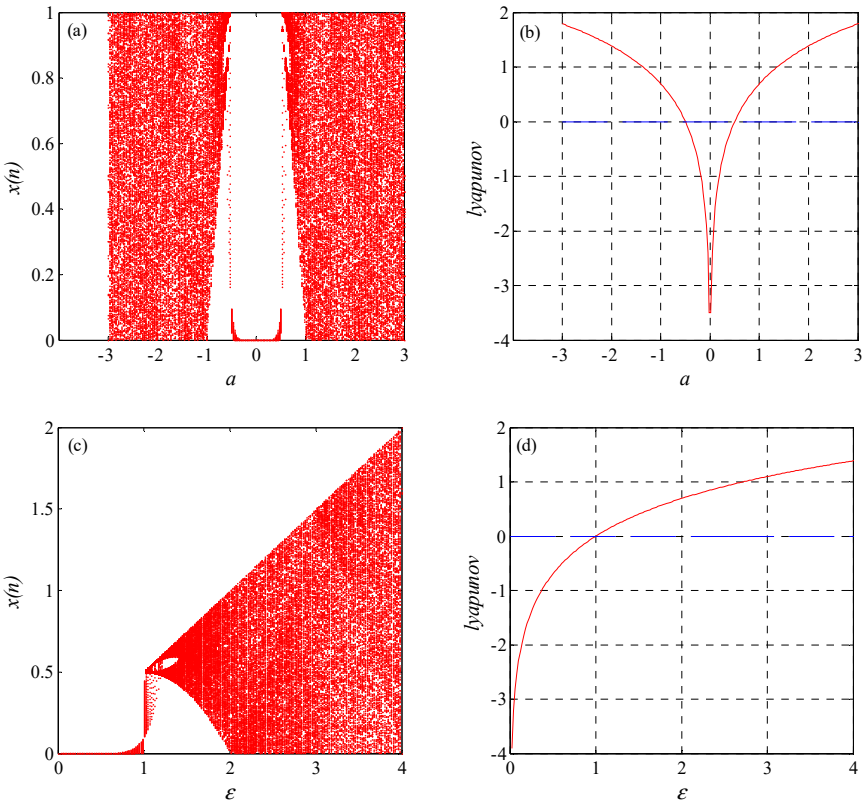


Figure 2. Bifurcation diagrams and Lyapunov exponent spectra of system (5). Let $\varepsilon = 2$: (**a**) bifurcation diagram of a and (**b**) Lyapunov exponent spectrum of a. Let $a = 1$: (**c**) bifurcation diagram of ε and (**d**) Lyapunov exponent spectrum of ε.

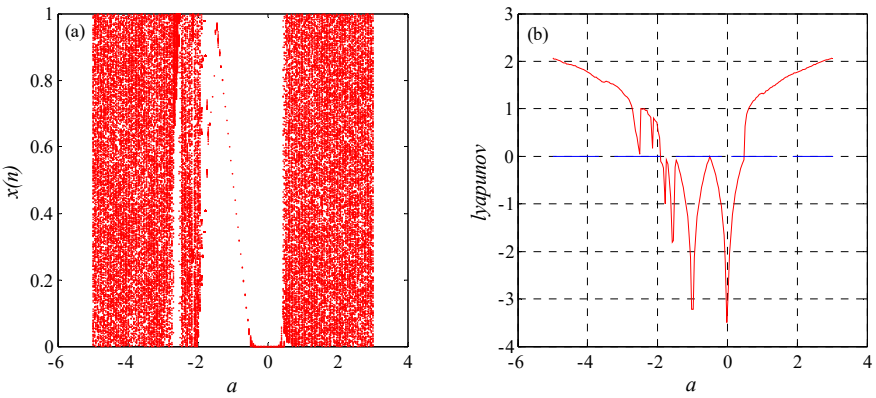


Figure 3. System (7): (**a**) bifurcation diagram of a and (**b**) Lyapunov exponent spectrum of a.

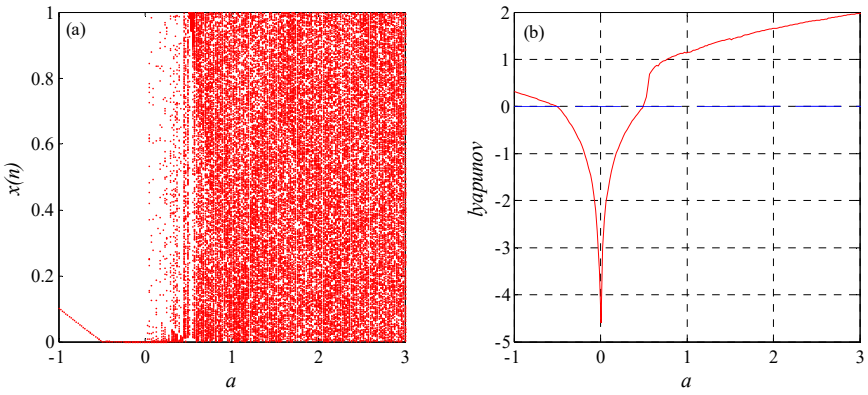


Figure 4. System (9): (**a**) bifurcation diagram of a and (**b**) Lyapunov exponent spectrum of a.

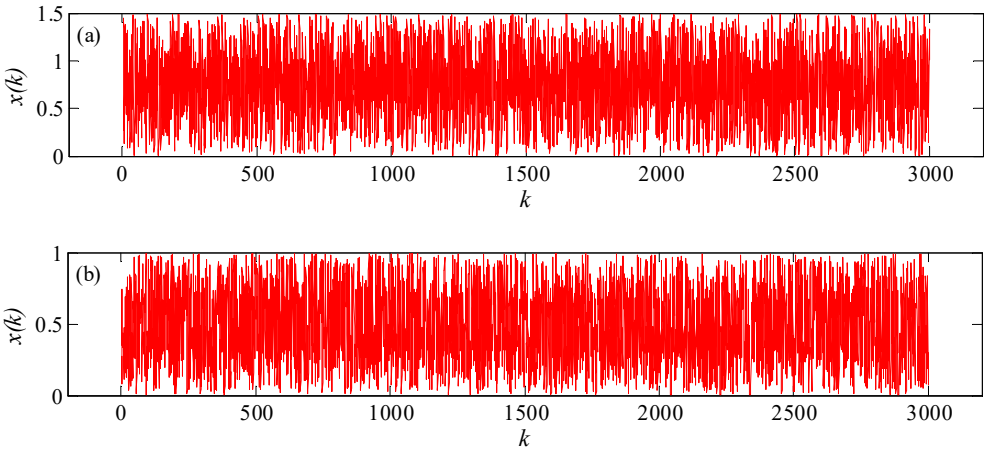


Figure 5. The evolution of the state variable: (**a**) system (5) and (**b**) system (7).

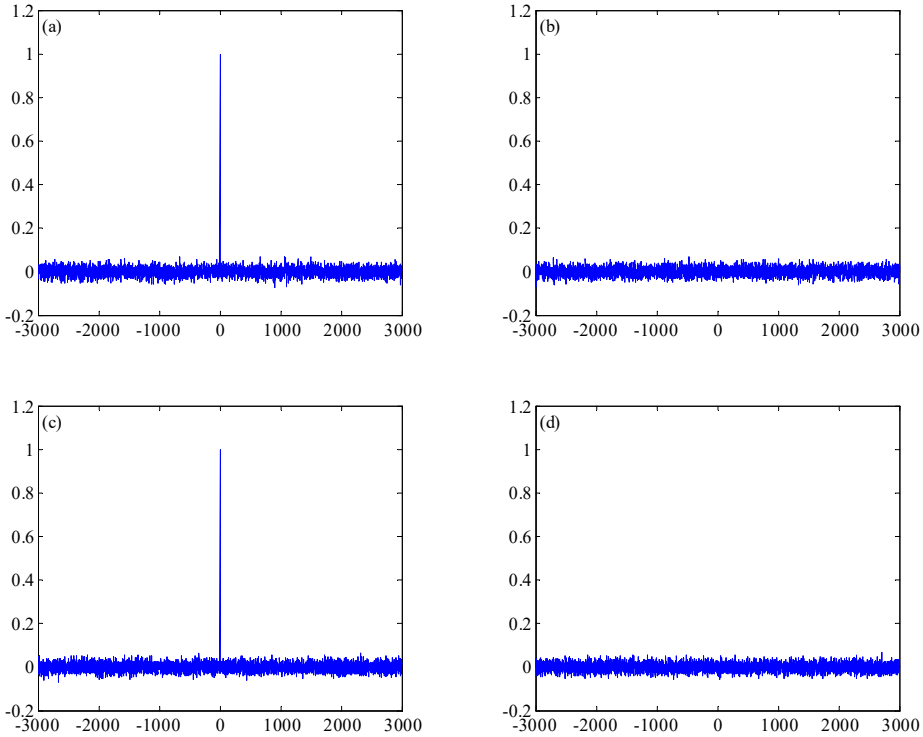


Figure 6. System (5): (**a**) autocorrelation function and (**b**) cross-correlation function. System (7): (**c**) autocorrelation function and (**d**) cross-correlation function.

For system (7), the autocorrelation function of chaotic sequence generated with initial parameter $a = 2$, $\varepsilon = 2$, $x_0 = 0.2759$ is shown in Figure 6c and its cross-correlation with another chaotic system generated with $a = 2$, $\varepsilon = 2$, $x_0 = 0.3257$ is shown in Figure 6d.

Figure 6 shows that the autocorrelation and cross-correlation functions of the chaotic systems generated by systems (5) and (7) are all ideal, as expected, which means that their pseudorandomness is very close to that of a truly random sequence.

2.3.3. Distribution Density Analysis

In practical applications, the distribution density of chaotic systems is usually required to be uniform or nearly uniform. In this section, the distribution density of chaotic systems based on Theorems 1 and 2 is investigated by means of histograms. First, the simulation method of the chaotic system distribution density is described as follows.

Step 1 First, $\{x(n)\}$ is denoted as a chaotic sequence of length N generated by a chaotic system. Assume the value range $\{x(n)\}$ of is $\Delta = [\alpha, \beta]$. In fact, $\alpha = \min\{x\{n\}\}$, $\beta = \max\{x\{n\}\}$.

Step 2 The interval Δ is divided into M subintervals equally, and the length of each subinterval is $h = (\beta - \alpha)/M$.

Step 3 The number of samples that fall into each subinterval is counted and denoted as n_i, $i = 1, 2, \cdots, M$.

Step 4 The probability of every point in the subinterval is denoted as p_i, $i = 1, 2, \cdots, M$. Then, probability p_i can be approximated by:

$$p_i = \frac{n_i}{N\Delta_i}.$$

Thus, $\sum_{i=1}^{N} p_i \Delta_i = 1$.

Then, the distribution density of a chaotic system can be simulated by the corresponding probability histogram in terms of the above method. In the following simulation, without specific declaration, $N = 10^6$, $M = 500$.

In Theorem 1, let $a = 1$; then, $\varepsilon = 3$, $\varepsilon = 3.3$, $\varepsilon = 3.8$, and $\varepsilon = 3.99$ are set, and histograms of the chaotic sequences generated by system (5) are shown in Figure 7a–d.

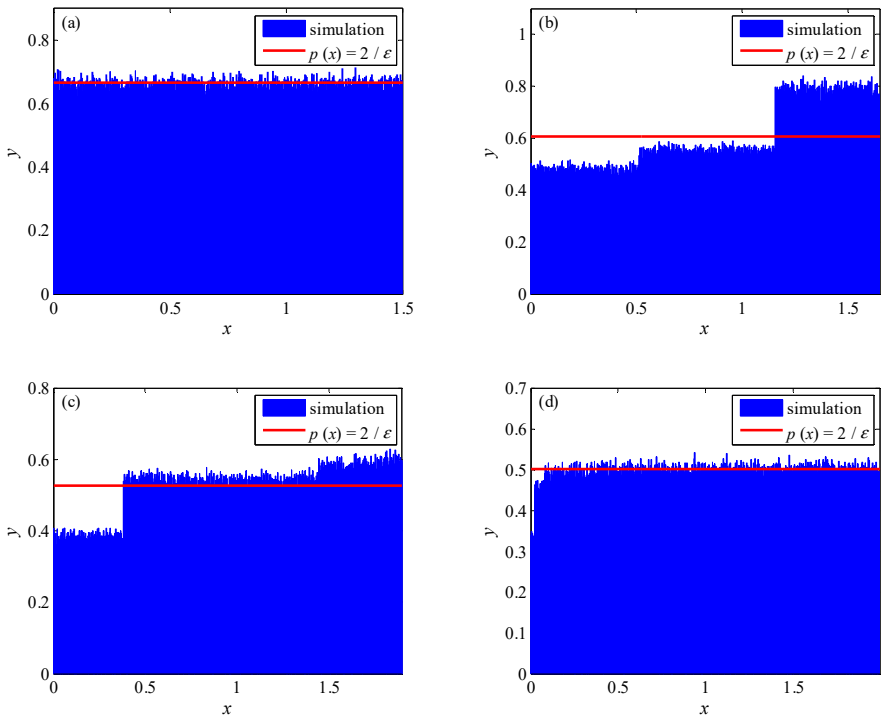


Figure 7. Histogram of system (5) with $a = 1$: (**a**) $\varepsilon = 3$; (**b**) $\varepsilon = 3.3$; (**c**) $\varepsilon = 3.8$; and (**d**) $\varepsilon = 3.99$.

In each histogram, the simulation represents the approximate value of probability p_i, and the curve $p(x) = 2/\varepsilon$ is used to fit the simulation of the distribution density of chaotic systems. Figure 7 shows that the distribution density of system (5) can be close to a uniform distribution by varying the system parameters, as shown in Figure 5a,d.

In fact, if set $a = 1$ and $\varepsilon = 2$, the linear system (5) is a tent map that follows a uniform distribution. It can be verified that if $|a\varepsilon| \geq 2$ is an integer, system (5) follows a uniform distribution in terms of the proof of the tent map. Now, the distribution density of nonlinear system (6) is studied, and for simplicity, system (7) is utilized as an example to study this problem.

Without loss of generality, we set $\varepsilon = 2$ in system (7) and then set $a = 1$, $a = 1.5$, $a = 1.8$, and $a = 2$, respectively, and histograms of the chaotic sequences generated by system (7) are shown in Figure 8a–d. As shown in Figure 8, the distribution density of system (7) can also be close to the uniform distribution by varying the system parameters.

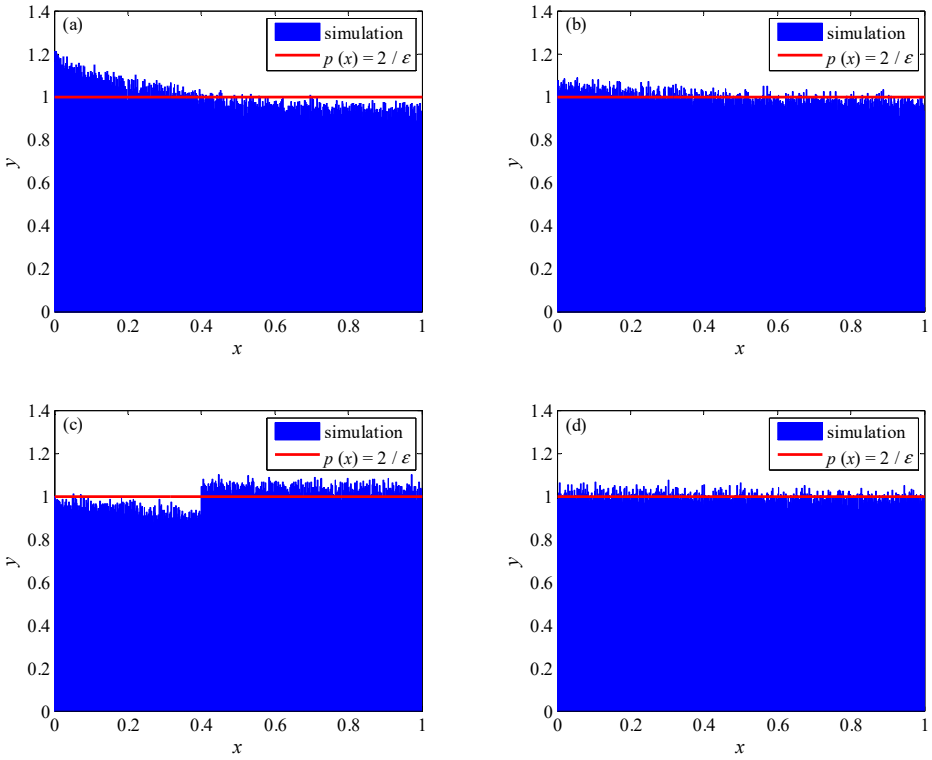


Figure 8. Histogram of system (7) where $\varepsilon = 2$: (**a**) $a = 1$; (**b**) $a = 1.5$; (**c**) $a = 1.8$; and (**d**) $a = 2$.

3. The Proposed Image Encryption Scheme

3.1. DNA Encoding and Computing Rules

A DNA sequence consists of four basic nucleic acids: A (adenine), C (cytosine), G (guanine), and T (thymine). According to the pair rules, A and T are complementary, as are C and G. In the binary system, 0 and 1 are complementary. Therefore, binary numbers 00 and 11, and 10 and 01 are also complementary. If we use the four basic nucleic acids (i.e., A, C, G, T) to denote the four binary numbers 00, 01, 10, and 11, there are in total 4! = 24 kinds of encoding rules. However, only eight rules which satisfy the Watson–Crick complementary requirement are valid. The rules are shown in Table 1.

According to the rules of DNA encoding and decoding, DNA sequences can be computed using algebraic calculation, such as addition, subtraction, and XOR operations. Table 2 lists the three operations for DNA sequences according to Rule 1.

Table 1. DNA encoding rules.

Rule	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
00	A	A	T	T	C	C	G	G
01	G	C	C	G	A	T	A	T
10	C	G	G	C	T	A	T	A
11	T	T	A	A	G	G	C	C

Table 2. DNA computing rules.

+	**A**	**G**	**C**	**T**
A	A	G	C	T
G	G	C	T	A
C	C	T	A	G
T	T	A	G	C
-	**A**	**G**	**C**	**T**
A	A	T	C	G
G	G	A	T	C
C	C	G	A	T
T	T	C	G	A
XOR	**A**	**G**	**C**	**T**
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

3.2. Iterations of Chaotic Systems

For chaotic system (5), we take three group parameters $\{a_1, x_{01}\}$, $\{a_2, x_{02}\}$, and $\{a_3, x_{03}\}$, and iterate $N_0 + MN$ times. In order to avoid the harmful effect of the transition procedure, we discard the previous N_0 sequence value and we obtain three chaotic sequences of length $M \times N$:

$$X_1 = \{x_1(i), i = 1, 2, \ldots, M \times N\}$$
$$X_2 = \{x_2(i), i = 1, 2, \ldots, M \times N\}$$
$$X_3 = \{x_3(i), i = 1, 2, \ldots, M \times N\}$$

The sequence values of the first group of chaotic sequences X_1 are sorted from small to large, and the corresponding subscript sequences are recorded to obtain an ordered subscript sequence $XP_1 = \{xp_1(i), i = 1, 2 \ldots M \times N\}$. For example, the sequence value of $xp_1(3)$ in XP_1 is 18, which means that the sequence number $x_1(3)$ of the sequence value in the chaotic sequence X_1 in the whole sequence X_1 is 18.

Similarly, the ordered subscript sequence XP_2 of chaotic sequence X_2 can be obtained.

The third group of chaotic sequences X_3 is transformed into an integer column r between [0, 255].

$$r = \mathrm{mod}\left(ceil\left(x_3 \times 10^8\right), 256\right)$$

The integer column r is arranged in rows into a matrix with size $M \times N$, which is recorded as R;

For chaotic system (8), two sets of parameters $\{b_1, b_2, b_3, b_4, b_5, b_6, x_{04}\}$, $\{c_1, c_2, c_3, c_4, c_5, c_6, x_{05},\}$ are taken and iterated $N_0 + MN$ times. Discarding the previous N_0 sequence value, we obtain two chaotic sequences of length $M \times N$:

$$X_4 = \{x_4(i), i = 1, 2, \ldots, M \times N\}$$
$$X_5 = \{x_5(i), i = 1, 2, \ldots, M \times N\}$$

The chaotic sequences X_4 and X_5 are transformed into two pseudorandom sequences between [0, 255]:

$$x_4(i) = \mathrm{mod}\left(floor\left(\frac{L(x_4(i) - \min(x_4))}{\max(x_4) - \min(x_4)}\right), 256\right)$$
$$x_5(i) = \mathrm{mod}\left(floor\left(\frac{L(x_5(i) - \min(x_5))}{\max(x_5) - \min(x_5)}\right), 256\right)$$

where $L = 255\sqrt{2} \times 10^8$ is a constant.

3.3. Proposed Image Encryption Scheme

Suppose the plain image is $P = (p(i, j))_{M \times N}, i = 1, 2 \ldots M, j = 1, 2 \ldots N$.

Step 1: Convert the image matrix P into a one-dimensional array and the pixel position of the image P is transformed by using the ordered subscript sequence XP_1 to obtain the image P_1.

Step 2: The image P_1 is divided into blocks and N pixels are taken in turn as a group to obtain M subimage $PR_i = \{pr_i(j), j = 1, 2, \ldots, N\}, i = 1, 2, \ldots, M$.

Step 3: Combined with chaotic sequence X_4, the pixel value of each subimage PR_i is transformed forward to obtain the subimage $PRN_i = \{prn_i(j), j = 1, 2, \ldots, N\}$. The specific transformation is:

$$prn_i(j) = pr_i(j) \oplus x_4((i-1)N + j) \oplus k_{i-1}, j = 1, 2, \ldots, N, i = 1, 2 \ldots M,$$
$$k_i = prn_i(1) \oplus prn_i(2) \oplus \ldots \oplus prn_i(N), i = 1, 2 \ldots M - 1,$$

where $\oplus$ represents XOR operation, $k_0 = \mathrm{mod}\left(\sum_i^M \sum_j^N pr_i(j) + i + j, 256\right)$ as a key.

Step 4: All pixel values in each subimage PRN_i are shifted to the left circularly in bits, and the moving bit is $d \in \{1, 2, \ldots, 8\}$ to obtain the subimage $PRB_i = \{prb_i(j), j = 1, 2, \ldots, N\}$. The specific transformation is:

$$prb_i(j) = \mathrm{mod}\left(prn_i(j) \times 2^d, 256\right) + floor\left(prn_i(j)/2^{8-d}\right)$$

Step 5: Combined with chaotic sequence X_5, the pixel value of each subimage PRB_i is inversely transformed to obtain a new subimage PRC_i. The specific transformation is:

$$prc_i(j) = prb_i(j) \oplus x_5((i-1)N + j) \oplus l_{i-1}, j = N, N-1, \ldots, 1, i = M, M-1 \ldots 1,$$
$$l_{i-2} = prc_i(1) \oplus prc_i(2) \oplus \ldots \oplus prc_i(N), i = M, M-1 \ldots 2,$$

where $l_{M-1} = \mathrm{mod}\left(\sum_i^M \sum_j^N prb_i(j) + i + j, 256\right)$ as a key.

Step 6: The PRC_i subimages are spliced to obtain the image P_2, and the pixel position of the image P_2 is transformed by the ordered subscript sequence XP_2. The transformed image P_3 is rearranged into an image of size $M \times N$.

Step 7: The image P_3 is encoded as a DNA matrix, and the matrix R in Section 3.2 is also encoded as a DNA matrix. The two DNA matrices are operated, the calculated DNA matrix is decoded to obtain a binary matrix, and finally, it is converted into a decimal matrix, that is, the last ciphertext image. The encoding and decoding rules and operation rules

in the step are determined by the plaintext image. The calculation formula of encoding, decoding, and operation rules is as follows:

$$\begin{aligned} r_b &= \mathrm{mod}(floor(sum(P) * 0.68), 8) + 1 \\ r_r &= \mathrm{mod}((sum(P) + i + j), 8) + 1 \\ r_y &= \mathrm{mod}(sum(P), 3) \\ r_j &= \mathrm{mod}(ceil(sum(P)/126), 8) + 1 \end{aligned}$$

where $sum(P)$ represents the sum of pixel values of image P, r_b represents the coding rules of image P_3, r_r represents the coding rules of matrix R, r_y represents the operation rules, and r_j represents the decoding rules.

The encryption process can be described by Figure 9.

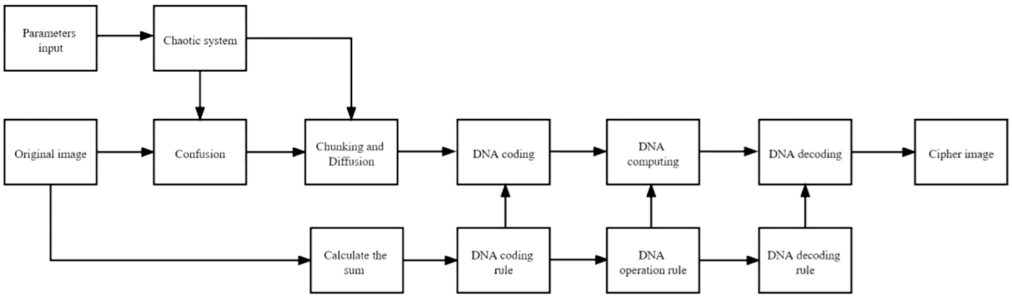


Figure 9. Image encryption scheme.

The decryption algorithm is the reverse process of the encryption algorithm.

4. Simulation Results and Security Analysis

We used MATLAB 2016a to verify the proposed encryption algorithm on a personal computer with an Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz and 8.00 GB memory, and the operating system was a Microsoft Windows 10. The test images are Lena (256×256), Girl(256×256), and Baboon(256×256). The system parameters of the chaotic system are

$$\left\{ \begin{array}{c} a_1 = 2, x_{01} = 0.2759, a_2 = 1.5, x_{02} = 0.3257, a_3 = 2, x_{03} = 0.28, \\ b_1 = 2, b_2 = b_3 = b_4 = b_5 = b_6 = 1, x_{04} = 0.6871, \\ c_1 = 3, c_2 = 2, c_3 = c_4 = c_5 = c_6 = 1, x_{05} = 0.4179 \\ n_0 = 1000, d = 3 \end{array} \right\}$$

The simulation results are shown in Figure 10.

4.1. Key Space Analysis

Key space measures the ability to resist exhaustive attacks. The key space size needs to be analyzed and calculated in combination with the system parameters involved in encryption, initial value conditions, and computer accuracy. Generally, the more key parameters there are, the greater the sensitivity of the key, the larger the key space, and the more difficult it is for the encryption and decryption algorithm to crack. In many other studies, the calculation accuracy is usually 10^{-14}. Therefore, this paper sets the calculation accuracy to 10^{-14} to compare with the key space of the same scale.

The key space of this algorithm is $\left(10^{14}\right)^{20} = 10^{280}$. Since the keys $\{n_0, k_0, l_{M-1}, d, e, f, g, h\}$ are numbers in the integer field, the key space is not taken into account when calculating. Therefore, the key space of this algorithm is much larger than the minimum value of resisting violent attacks and the key space in the following references. The comparison with the key space in other references is provided in Table 3.

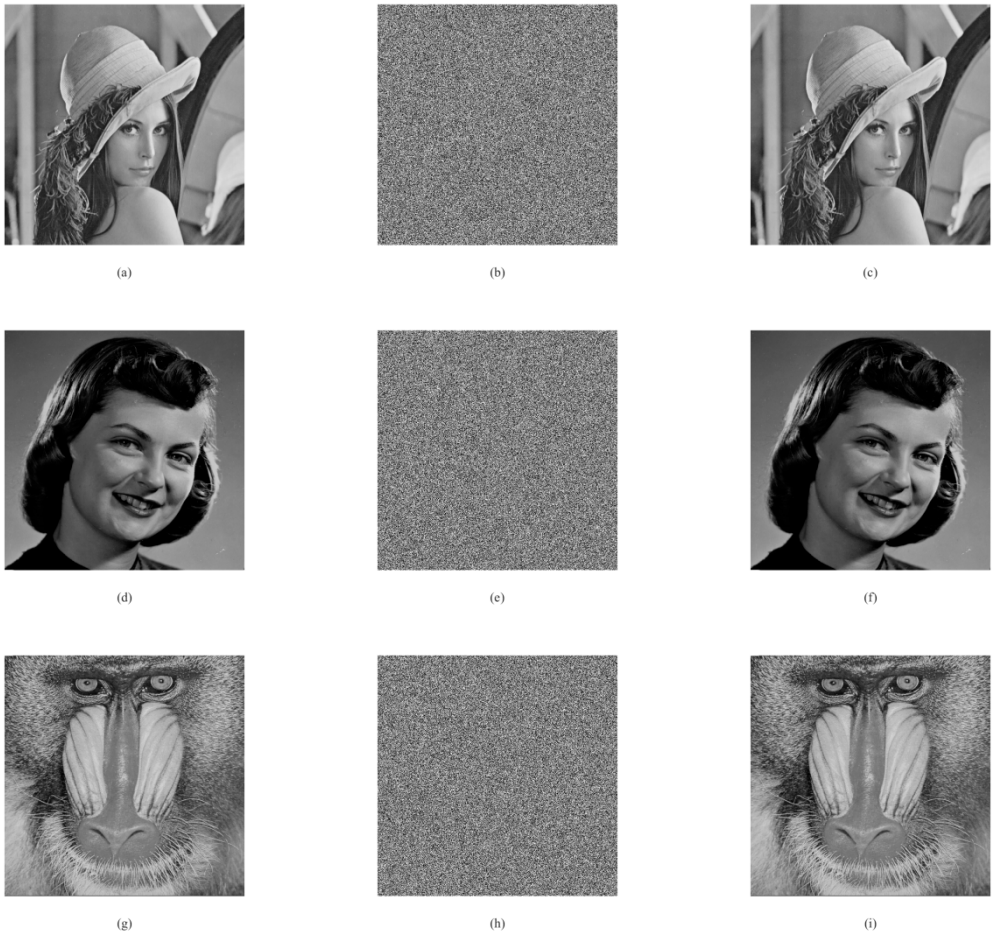


Figure 10. Images encryption and decryption effect: (**a**,**d**,**g**) original image; (**b**,**e**,**h**) encrypted image; and (**c**,**f**,**i**) decrypted image.

Table 3. Key space analysis.

	Ref. [23]	**Ref. [24]**	**Ref. [25]**	**Ref. [26]**	**Proposed**
Key space	10^{93}	10^{98}	10^{84}	10^{142}	10^{280}

4.2. Key Sensitivity Analysis

A good encryption scheme should be sensitive to the key in the encryption process. Below, the key is slightly disturbed, and NPCR and UACI are used to measure the differences of the images before and after the perturbation.

The NPCR and UACI are expressed as follows:

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H} D(i,j)}{W \times H} \times 100\%,$$

$$\mathrm{UACI} = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H} \frac{|p_1(i,j)-p_2(i,j)|}{L-1}}{W \times H} \times 100\%,$$

$$D(i,j) = \begin{cases} 0, p_1(i,j) = p_2(i,j) \\ 1, p_1(i,j) \neq p_2(i,j) \end{cases},$$

where $W \times H$ is the number of pixels.

The key sensitivity analysis results of the encryption algorithm are shown in Table 4. The values of NPCR and UACI are very close to the ideal values of 99.609% and 33.464%, respectively, after a minor disturbance of the parameters. Therefore, there are large differences in decrypted images and high key sensitivity.

Table 4. Key sensitivity analysis.

Initial Parameters	Minor Disturbance	NPCR	UACI
a_1	$+10^{-14}$	99.63%	33.51%
x_{01}	$+10^{-14}$	99.61%	33.45%
a_2	$+10^{-14}$	99.62%	33.49%
x_{02}	$+10^{-14}$	99.60%	33.52%
a_3	$+10^{-14}$	99.60%	33.45%
x_{03}	$+10^{-14}$	99.62%	33.46%
b_1	$+10^{-14}$	99.59%	33.47%
b_2	$+10^{-14}$	99.62%	33.47%
b_3	$+10^{-14}$	99.60%	33.44%
b_4	$+10^{-14}$	99.63%	33.47%
b_5	$+10^{-14}$	99.61%	33.45%
b_6	$+10^{-14}$	99.61%	33.38%
x_{04}	$+10^{-14}$	99.61%	33.40%
c_1	$+10^{-14}$	99.58%	33.47%
c_2	$+10^{-14}$	99.60%	33.39%
c_3	$+10^{-14}$	99.62%	33.45%
c_4	$+10^{-14}$	99.59%	33.45%
c_5	$+10^{-14}$	99.62%	33.44%
c_6	$+10^{-14}$	99.60%	33.45%
x_{05}	$+10^{-14}$	99.61%	33.48%

4.3. Histogram Analysis

The statistical histogram directly observes the encryption effect of image encryption and reflects the distribution of pixels by comparing the pixel statistical histogram of the original image with the encrypted image. It is generally considered that the statistical histogram of an encrypted image is approximately uniform. Figure 11 shows the encrypted histogram of the Lena image. The histogram distribution is flat and better hides the statistical law of pixels. It can effectively resist statistical attacks and pure password attacks.

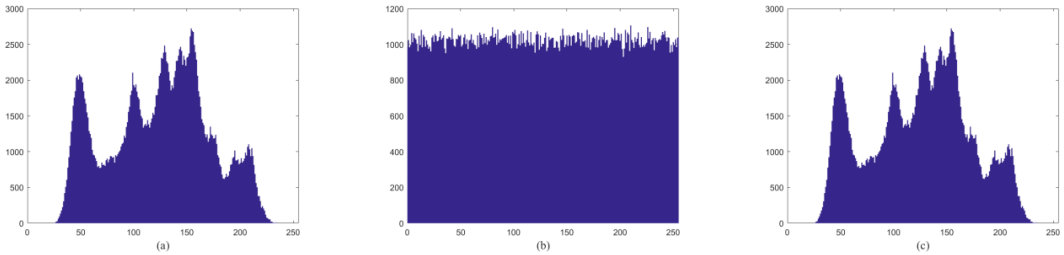


Figure 11. The result of Lena histogram analysis: (**a**) original image; (**b**) encrypted image; and (**c**) decrypted image.

4.4. Correlation Analysis

The correlation coefficient calculates the correlation coefficient of the original image and the encrypted image, compares the absolute value of the correlation coefficient, judges the correlation change of adjacent pixels of the encrypted image, and measures the correlation of adjacent pixels.

In digital images, the gray values between adjacent pixels are often very close, indicating that adjacent pixels have a strong correlation, which will lead to insufficient encryption security performance. When the absolute value of the correlation coefficient is close to

1, it is considered that there is a strong correlation between adjacent pixels; when the absolute value of the correlation coefficient is close to 0, it is considered that there is no or weak correlation between adjacent pixels. The calculation of the correlation coefficient between adjacent pixels is divided into three directions: vertical, horizontal, and diagonal. Equations (10)–(12) are used to calculate the correlation between adjacent pixels of an image:

$$Cov(x, y) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)][y_i - E(y)], \tag{10}$$

$$D(x) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2, \tag{11}$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n}x_i, \tag{12}$$

where x and y are the gray values of two adjacent pixels. $E(x)$, $D(x)$, and $Cov(x, y)$ are expectation, variance, and covariance. In this paper we select 2000 pairs of pixels in Lena's plain image and cipher image.

The relevant distributions of adjacent pixels of the Lena image along the horizontal, vertical, and diagonal directions are shown in Figure 12a–c, respectively, and the corresponding distributions of encrypted images are shown in Figure 12d–f. The results show that the correlation between adjacent pixels in ordinary images is greatly reduced. The comparison results with other studies are shown in Table 5.

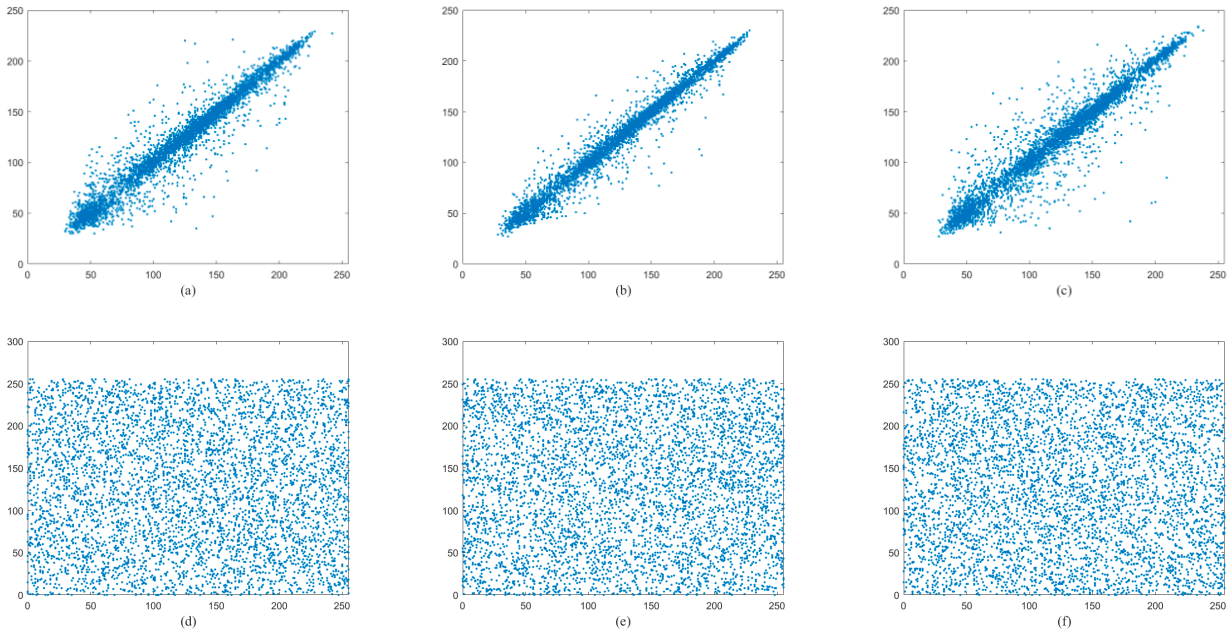


Figure 12. Correlation analysis of the Lena image. (**a**) Horizontal direction; (**b**) vertical direction; and (**c**) diagonal direction. Encrypted image of (**d**) horizontal direction; (**e**) vertical direction; and (**f**) diagonal direction.

4.5. Information Entropy Analysis

Information entropy is used to judge the degree of unpredictability, uncertainty, and randomness of information sources. Information entropy is expressed as Equation (13):

$$H(S) = -\sum_{i=1}^{n} p_i \log p_i, \tag{13}$$

where $S = \{x_1, x_2, \ldots x_n\}$ is an information source and P is a probability distribution of S. The probability of x_i is p_i. According to the principle of maximum information entropy, when the probability distribution of the source is an equal probability distribution, $p_i = \frac{1}{n}$, and the maximum information entropy $\log_2 n$ can be obtained.

Table 5. Correlation analysis.

Correlation	Horizontal	Vertical	Diagonal
Lena	0.9849	0.9704	0.9611
Ref. [23]	0.0007	0.0015	0.0014
Ref. [24]	-	-	-
Ref. [25]	−0.0034	−0.0079	0.0010
Ref. [26]	0.001	−0.014	−0.006
Proposed	0.0072	0.0055	−0.0008

Since the image used in this experiment is a 256-order gray image, the closer the information entropy is to 8, the better the encryption algorithm performs in this test. The original picture information entropy = 7.4455, and the picture information entropy encrypted by the algorithm in this paper = 7.9994, which is very close to the ideal value of 8. Table 6 shows the comparison of entropy with other literature.

Table 6. Information entropy analysis.

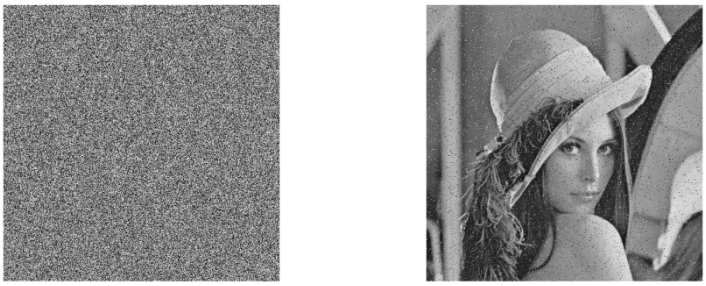
	Ref. [23]	Ref. [24]	Ref. [25]	Ref. [26]	Proposed
Information entropy	7.9967	7.95667	7.9977	7.9994	7.9994

4.6. Robustness Analysis

In the process of image transmission, some data may be modified or lost. Therefore, an algorithm should have the ability to resist noise attacks or data loss. A robust encryption algorithm means that most of the useful information of the plain image can still be recovered when such situations occur.

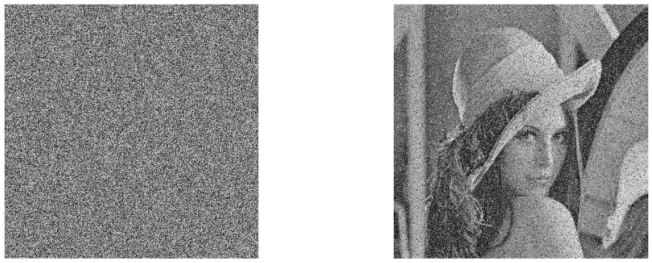
Currently, almost all transmission channels are noise channels. When data propagate in the channel, it receives various types of noise interference, such as Gaussian noise and salt and pepper noise. A robust image encryption algorithm should be immune to noise interference. In the actual algorithm analysis, a small amount of some type of noise is usually added to the encrypted image and then the encrypted image is decrypted after adding noise. The smaller the contrast difference between the decrypted image and the original image, the stronger the ability of the algorithm to resist noise attacks.

We added 0.01%, 0.03%, and 0.05% salt and pepper noise. The results are shown in Figures 13–15, respectively. These figures show that the decryption algorithm can still restore the original image well, that is, it has a certain ability to resist noise attacks.



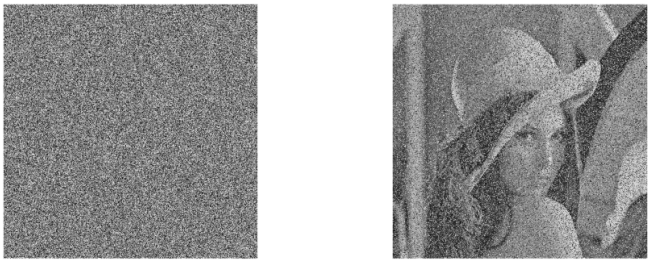
(a) (b)

Figure 13. Salt and pepper noise level of 0.01%: (**a**) encrypted image and (**b**) decrypted image.



(a) (b)

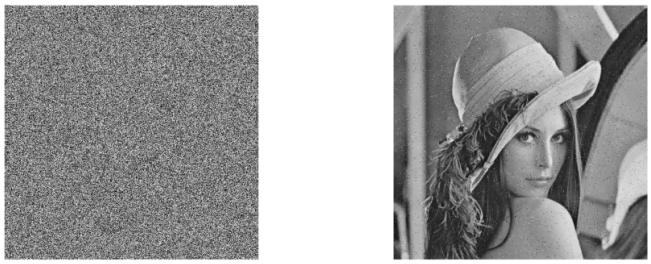
Figure 14. Salt and pepper noise level of 0.03%: (**a**) encrypted image and (**b**) decrypted image.



(a) (b)

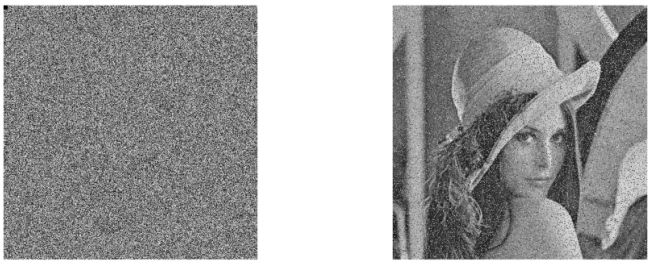
Figure 15. Salt and pepper noise level of 0.05%: (**a**) encrypted image and (**b**) decrypted image.

During transmission, the encrypted data may be partially modified or lost. An encryption algorithm should be immune to data loss. In the actual algorithm analysis, a small image of the encrypted image is usually removed, and then the encrypted image after removing part of the image is decrypted. The smaller the contrast difference between the decrypted image and the original image, the stronger the ability of the algorithm to resist data loss attacks. The 1×8, 8×8, 8×16 subblock in the upper left corner of the encrypted image is removed in Figure 10b. The decryption algorithm is used to decrypt the encrypted image after removing a molecular block. The decryption effect is shown in Figures 16b, 17b, 18b. Figures 16–18 show that the decryption algorithm can still restore the original image well; that is, it has a certain ability to resist data loss.



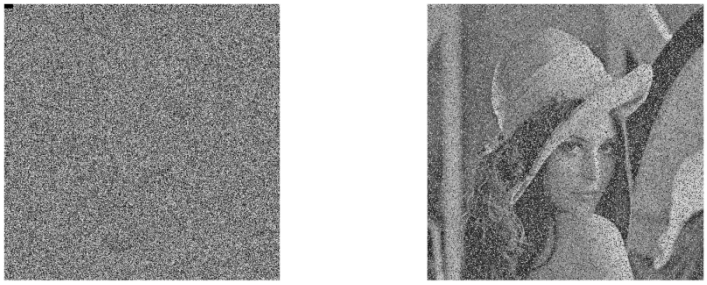
(a) (b)

Figure 16. The removal of a 1×8 subblock: (**a**) encrypted image and (**b**) decrypted image.



(a) (b)

Figure 17. The removal of an 8×8 sub block: (**a**) encrypted image and (**b**) decrypted image.



(a) (b)

Figure 18. The removal of an 8×16 sub block: (**a**) encrypted image and (**b**) decrypted image.

4.7. Time Complexity Analysis

The time consumed by the encryption algorithm and decryption algorithm provided in this paper can be divided into two stages: (1) preparation stage, that is, the generation of chaotic pseudorandom sequences; and (2) formal encryption and decryption stage. The time of phase (1) in this algorithm is 8.948 s. For phase (2), the time consumed by the encryption algorithm and decryption algorithm is 3.702 s and 3.893 s, respectively. Therefore, the chaotic image encryption algorithm in this paper consumes less time; that is, its time complexity is low.

5. Conclusions

This paper proposes a method to construct a one-dimensional discrete chaotic system and an image encryption scheme based on a uniformly distributed chaotic system. Based on Marotto's theorem, one-dimensional discrete systems are proven to be chaotic in the sense of Li–Yorke, and the corresponding chaos criterion theorems are proposed. The system can be distributed uniformly which means better randomness. We propose an image encryption scheme based on a uniformly distributed discrete chaotic system and DNA encoding. The experimental results demonstrate that our encryption algorithm has a large key space, high key sensitivity, and fast encryption speed and can resist differential attacks and statistical attacks.

Author Contributions: H.Z. carried out numerical simulation and analysis; M.T. and X.W. studied the proof of relevant theories and proposed the image encryption scheme; H.Z. and M.T. wrote the paper; and H.Z. and M.T. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Tutueva, A.V.; Nepomuceno, E.G.; Karimov, A.I.; Andreev, V.S.; Butusov, D.N. Adaptive chaotic maps and their application to pseudo-random numbers generation. *Chaos Solitons Fractals* **2020**, *133*, 109615. [CrossRef]
2. Wang, L.Y.; Cheng, H. Pseudo-Random Number Generator Based on Logistic Chaotic System. *Entropy* **2019**, *21*, 960. [CrossRef]
3. Lambic, D. A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. *Nonlinear Dyn.* **2020**, *100*, 699–711. [CrossRef]
4. Jiang, D.H.; Liu, L.D.; Zhu, L.Y.; Wang, X.Y.; Rong, X.W.; Chai, H.X. Adaptive embedding: A novel meaningful image encryption scheme based on parallel compressive sensing and slant transform. *Signal Process.* **2021**, *188*, 108220. [CrossRef]
5. Cheng, G.F.; Wang, C.H.; Chen, H. A Novel Color Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture. *Int. J. Bifurc. Chaos* **2019**, *29*, 1950115. [CrossRef]
6. Chai, X.L.; Fu, X.L.; Gan, Z.H.; Lu, Y.; Chen, Y.R. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [CrossRef]

7. Wang, J.; Zhi, X.C.; Chai, X.L.; Lu, Y. Chaos-based image encryption strategy based on random number embedding and DNA-level self-adaptive permutation and diffusion. *Multimed. Tools Appl.* **2021**, *80*, 16087–16122. [CrossRef]
8. Wang, X.Y.; Zhao, H.Y.; Hou, Y.T.; Luo, C.; Zhang, Y.Q.; Wang, C.P. Chaotic image encryption algorithm based on pseudo-random bit sequence and DNA plane. *Mod. Phys. Lett. B* **2019**, *33*, 1950263. [CrossRef]
9. Zang, H.; Zhao, X.; Wei, X. Construction and application of new high-order polynomial chaotic maps. *Nonlinear Dyn.* **2022**, *107*, 1247–1261. [CrossRef]
10. Wei, X.Y.; Zang, H.Y. Construction and complexity analysis of new cubic chaotic maps based on spectral entropy algorithm. *J. Intell. Fuzzy Syst.* **2019**, *37*, 4547–4555. [CrossRef]
11. Li, T.Y.; Yorke, J.A. Period Three Implies Chaos. *Am. Math. Mon.* **1975**, *82*, 985–992. [CrossRef]
12. Marotto, F.R. Snap-back repellers imply chaos in Rn. *J. Math. Anal. Appl.* **1978**, *63*, 199–223. [CrossRef]
13. Chen, G.; Hsu, S.; Zhou, J. Snapback repellers as a cause of chaotic vibration of the wave equation with a van der Pol boundary condition and energy injection at the middle of the span. *J. Math. Phys.* **1998**, *39*, 6459–6489. [CrossRef]
14. Zhou, H.L.; Song, E.B. Discrimination of the 3-periodic points of a quadratic polynomial. *J. Sichuan Univ.* **2009**, *46*, 561–564.
15. Yang, X.P.; Min, L.Q.; Wang, X. A cubic map chaos criterion theorem with applications in generalized synchronization based pseudorandom number generator and image encryption. *Chaos* **2015**, *25*, 053104. [CrossRef] [PubMed]
16. Chen, G.R.; Lai, D.J. Feedback control of Lyapunov exponent for discrete-time dynamical systems. *Int. J. Bifurc. Chaos* **1996**, *6*, 1341–1349. [CrossRef]
17. Yu, S.M.; Lv, J.H.; Chen, G.R. *Anti-Control Method of Dynamical Systems and Its Application*, 2nd ed.; Science Press: Beijing, China, 2013.
18. Zang, H.Y.; Li, J.; Li, G.D. A One-dimensional Discrete Map Chaos Criterion Theorem with Applications in Pseudo-random Number Generator. *J. Electron. Inf. Technol.* **2018**, *40*, 1992–1997.
19. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science* **1994**, *266*, 1021–1024. [CrossRef]
20. Kang, X.J.; Guo, Z.H. A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670.
21. Liu, Z.T.; Wu, C.X.; Wang, J.; Hu, Y.H. A Color Image Encryption Using Dynamic DNA and 4-D Memristive Hyper-Chaos. *IEEE Access* **2019**, *7*, 78367–78378. [CrossRef]
22. Liu, Q.; Liu, L.F. Color Image Encryption Algorithm Based on DNA Coding and Double Chaos System. *IEEE Access* **2020**, *8*, 83596–83610. [CrossRef]
23. Song, C.Y.; Qiao, Y.L. A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. *Entropy* **2015**, *17*, 6954–6968. [CrossRef]
24. Cavusoglu, U.; Kacar, S.; Pehlivan, I.; Zengin, A. Secure image encryption algorithm design using a novel chaos based S-Box. *Chaos Solitons Fractals* **2017**, *95*, 92–101. [CrossRef]
25. Zhang, S.J.; Liu, L.F.; Xiang, H.Y. A Novel Plain-Text Related Image Encryption Algorithm Based on LB Compound Chaotic Map. *Mathematics* **2021**, *9*, 2778. [CrossRef]
26. Nkandeu, Y.P.K.; Tiedeu, A. An image encryption algorithm based on substitution technique and chaos mixing. *Multimed. Tools Appl.* **2019**, *78*, 10013–10034. [CrossRef]