Research Article

An Image Encryption Scheme Using a 1D Chaotic Double Section Skew Tent Map

Rania A. Elmanfaloty and Ehab Abou-Bakr

1Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
2Department of Electronics and Communications Engineering, Alexandria Higher Institute of Engineering and Technology, Alexandria, Egypt
3Department of Electrical Engineering, College of Engineering and IT, Onaizah Colleges, Al-Qassim 56447, Saudi Arabia
4Department of Computer Engineering, The Higher Institute of Engineering and Technology, El-Behera, Egypt

Correspondence should be addressed to Rania A. Elmanfaloty; relmanfaloty@kau.edu.sa

Copyright © 2020 Rania A. Elmanfaloty and Ehab Abou-Bakr. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because of their complex behaviour, simple mathematical and digital hardware representation, one-dimensional (1D) chaotic functions were the earliest choices of cryptologists for applying chaos theory to cryptography. It was discovered later that most of these functions suffer from orbits collapsing into a specific period, especially under finite precision realization and weakness because of their limited number of control parameter(s). This in turn exposed many security issues and was proven to be vulnerable to various types of attacks. This paper addresses the issue of limited number of control parameters by introducing a 1D chaotic function with five control parameters (in addition to the initial condition). Analysis of the function implies its chaotic properties in addition to its ability to generate a cryptographically secured random stream of numbers. To further elaborate on its robustness, a new image encryption algorithm incorporating the function as its random number generator is presented. Various analyses of the new scheme confirm it to be secure with good confusion-diffusion properties.

1. Introduction

In the current era, digital technology is changing all aspects of everyday life. It has revolutionized the way data is stored, displayed, and transmitted. However, with the increase of digital-based applications such as mobile communications, cloud storage, and the Internet of things (IoT), more advanced methods of securing these data are also being developed. While many logical/physical techniques exist to protect sensitive data from unauthorized access and accidental or intentional destruction [1], encryption is the most popular method of choice for securing them. As depicted in Figure 1, an encryption/decryption should incorporate three basic elements: a message, an encryption/decryption scheme, and a key [2].

Despite the fact that the strength of an algorithm comes from the strength of its key [3, 4], other critical factors come into consideration for the overall performance. Some of these factors would be the computational load of their mathematical representation [5–11], the generated key size, and the randomness of the generated sequence.

This is one of many reasons why cryptologists have turned to chaotic functions as many chaotic functions are simple in their mathematical representation [12–14], such as 1-D functions with iterative difference equation [15]. These functions are also characterized by their high sensitivity to
initial conditions, aperiodicity, and unpredictability [16]. For all these reasons, these functions seem to be ideal for encryption application.

When it comes to hardware implementation, the cost of realizing low dimension maps (specially 1D maps) decreases significantly with respect to higher-order ones. However, PRNG with a single 1D chaotic map as its core proved to be inefficient with generally two main issues [17–22]; (i) its limited number of control parameters and (ii) its orbit collapsing to a specific period under finite precision implementation. Recent studies suggested that 2D chaotic maps may be considered as a midpoint balance between hardware complexity and chaotic performance. This is clearly shown in the recent work of Zhongyun Hua where the complexity of 2D chaotic maps was improved by using a two-dimensional sine chaotification system (2D-SCS) in [23] and two-dimensional modular chaotification system (2D-MCS) in [24].

For the second issue, [25, 26] investigated the effect of finite precision on the periodic and chaotic behaviour of the logistic and coupled logistic maps. In [27], further studies of the same effect were conducted on single and coupled skew tent maps, the authors then presented a modified design to reduce the effect of limited finite precision realization on coupled skew tent maps. The finding of the research gives deep insight on how this issue needs a lot of attention in hardware implementation of chaotic systems.

This manuscript deals with the first issue mentioned above by introducing a 1D difference chaotic map with five variable parameters (not including the initial condition). The map is first analyzed for chaotic properties and its capability to be used as pseudorandom number generator (PRNG). Then, based on the promising results of the map histograms and statistical analysis, the proposed map will be used as a RING in a new image encryption scheme. Despite the map simplicity, its unique feature of increasing the number of control parameters allowed a key space large enough to withstand brute-force attack. Further analysis of the whole encryption scheme showed good confusion-diffusion properties.

This paper is organized as follows: Section 2 gives the mathematical representation of the presented map with an investigation of its chaotic properties; in Section 3, a new image encryption algorithm using the introduced map is presented and the encryption results were subjected to many statistical tests to prove its robustness. The conclusion is given in Section 5.

2. A 1D Map with Variable Control Parameters

In this section, a 1D map with variable control parameters is introduced. Multiple analysis is conducted to validate its chaotic property. Then, the possibility of employing it as a potential pseudorandom bit generator (PRBG) is tested.

2.1. Mathematical Representation. The first order discrete dynamical system in Figure 2 is given by

\[
x_{n+1} = f(x) = \begin{cases} 
\frac{h}{p_1} x_n, & x \in [0, p_1], \\
\frac{h}{L_1 - p_1} (L_1 - x_n), & x \in (p_1, L_1], \\
\frac{h}{p_2 - L_1} (x_n - L_1), & x \in (L_1, p_2], \\
\frac{h}{L_2 - p_2} (L_2 - x_n), & x \in (p_2, L_2].
\end{cases}
\]

where \( x \in \mathbb{R}^+; x \in [0, L_2] \), and \( \{L_1, L_2, p_1, p_2, h\} \in \mathbb{R}^+ \) acting as the control parameters. Although the previous postulate stated the range of \( x \) and the control parameters to be in positive real number set \( \mathbb{R}^+ \), for the function to exhibit full-chaotic behaviour, the following conditions must be satisfied:

(i) \( L_1 < L_2 \),
(ii) \( p_1 \in (0, L_1) \),
(iii) \( p_2 \in (L_1, L_2) \),
(iv) \( h = L_2 \).

Although the function in (1) may seem to resemble the second iterate of the skew tent map given by

\[
f(y) = y_{n+1} = \begin{cases} 
\frac{y_n}{p}, & y \in \mathbb{R}; y \in (0, p], \\
1 - \frac{y_n}{1 - p}, & y \in \mathbb{R}; y \in (p, 1).
\end{cases}
\]

It is obvious that more system parameters were added to increase the control on its topology, where the map described by (2) has only one control parameter \( p \), while the proposed function has five \( \{L_1, L_2, p_1, p_2, h\} \). In addition, expanding the phase space to \( [0, L_2] \) makes it difficult to estimate the control parameters using chaotic signal estimation technologies for 1D maps [28, 29].
2.2. Chaotic Properties of the Proposed Function. In 1989, Devaney [30] set the most common definition for classifying dynamical systems as chaotic; that is, the system should (i) be topologically transitive, (ii) have dense periodic orbits, and (iii) be sensitive to the initial condition \((x_0)\). We follow this definition to validate the presented function chaotic properties. In addition to that, a numerical calculation of the Lyapunov exponent for different values of the control parameters was conducted.

2.2.1. Topologically Transitivity (Mixing). Letting \(X\) be a topological space, a continuous map \(f: X \rightarrow X\) is said to be topologically transitive on \(X\) if for any two open non-empty subintervals \(U_1, U_2 \subset X\), there exists an \(n > 0\) such that \(f^n(U_1) \cap U_2 \neq \emptyset\), where \(\emptyset\) is an empty set.

The first five iterates \(f^1, f^2, f^3, f^4, f^5\) of the function \(f: [0, L_2] \rightarrow [0, L_2]\) in (1) are shown in Figure 3. It is visually clear that any \(f^n\) iterate has \(k = 2^{2n-1}\) “skew tent” and the iterate \(f^n\) maps itself to the interval \([0, L_2]\). This implies that \(n\) satisfies \(f^n(U_1) \cap U_2 \neq \emptyset\) and that the presented function is topologically transitive.

2.2.2. Dense Periodic Orbits. By definition, a point \(x \in X\) is a periodic point for \(f\) with period \(n \in \mathbb{N}\) if \(f^n(x) = x\). Thus, by referring to Figure 3, it is obvious that \(f^n\) intersects a line \(y = x\) in \(2^{2n}\) locations (once in each interval). As a result, each interval contains a fixed point of \(f^n\) that results in a periodic orbit of period \(n\). Therefore, periodic points of \(f\) are dense in \([0, L_2]\).

2.2.3. Sensitivity to \(x_o, p_1, p_2, L_1, L_2\)

Sensitivity to the initial condition \(x_o\)

The map \(f: X \rightarrow X\) is sensitive to the initial conditions if for any \(x \in X\) and any neighbourhood \(N\) of \(x\), there exist \(y \in N\) and \(n \geq 0\) such that \(|f^n(x) - f^n(y)| > \delta\), where \(\delta > 0\).

The presented map shows high sensitivity to the initial condition as depicted in Figure 4. Two trajectories starting from two nearby initial points \([x_o, x_o + d]\) (where \(d \sim 10^{-9}\)) coincide at first and then evolve incoherently afterwards.

Sensitivity to the control parameters \(\{p_1, p_2, L_1, L_2\}\): To test the sensitivity of the proposed map for any small perturbation in the control parameters, a sequence \(S_1\) is generated using the values of \(\{p_1, p_2, L_1, L_2\}\). Then, another sequence \(S_2\) is generated but with a small perturbation of \(d \sim 10^{-9}\) to any of the control parameters. The cross-correlation results in Figure 5 show that no correlation was detected between \(S_1\) and the sequences of \(S_2\). This proves the sensitivity of the proposed map to any small change in any of the control parameters.

2.2.4. Lyapunov Exponent (LE). For a dynamical system with two infinitesimally close initial points \((x_0, x_0 + \delta)\) generating two nearby trajectories, the rate of exponential divergence of these nearby trajectories is called the Lyapunov exponent and is defined by

\[
\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{\delta_1}{\delta_o} \right| \tag{3}
\]

where \(\delta_o\) and \(\delta_1\) represent a separation between two trajectories. By itself, the Lyapunov exponent defines the degree of sensitivity to the initial condition. However, for a dynamical system bounded in a finite phase space, it is the familiar test for chaotic behaviour where positive LE indicates chaos.

The procedural steps described by Sprott in [31] allow the calculation of the Lyapunov exponent numerically as follows:

1. Choose any initial condition in the basin of the trajectory.
2. Sufficiently iterate the map to ensure it is on the trajectory. Select a nearby point with infinitesimal separation \(\delta_o\).
3. Increment both orbits on iteration and then calculate the new separation \(\delta_1 = |x_n - x_\delta_i|\).
4. Evaluate \(\ln \left| \left( \frac{\delta_1}{\delta_o} \right) \right|\).
5. Redo trajectory so its separation from the original trajectory is back to \(\delta_o\) and is in the same direction as \(\delta_1\) using \(x_{\text{o}, n} = x_{\text{o}, i} + \delta_o \frac{1}{\delta_1} (x_{\text{o}, i} - x_{\text{o}, i})\).
6. Repeat steps 4–6 multiple times and then calculate the average of step 5.

Since there are multiple system parameters controlling the topology of the presented function, Algorithm 1 shows how the Lyapunov exponent was calculated by varying \(p_1\) and \(p_2\) while fixing \(L_1\) to half of \(L_2\). The 3D depiction and contour representation of the result in Figures 6(a) and 6(b) show that, for all values of \(p_1\) and \(p_2\), \(\lambda > 0\) and subsequently the orbit is “unstable.” This emphasizes that the map possesses a chaotic behaviour for all values of \(p_1\) and \(p_2\). This is also considered an advantage over the LE of the tent map given by (4) and having only one control parameter \(\mu \in [0, 2]\). In comparison, it is visible from Figure 6(c) that, for the tent map to be “unstable” (\(\lambda > 0\)), \(\mu\) must be greater than “1,” whereas in the proposed map \(\lambda > 0\) for all values of the control parameters.
2.3. Cryptographic Property of the Presented 1D Map. The histogram count shown in Figure 7 reveals the uniformity of the output sequence for different (random) values of the control parameters \( p_1, p_2, L_1, L_2 \). This is an indication of good cryptographic property as a uniform distribution gives little information about the system and its control parameters.

The chaotic map described by (1) is used to generate a binary sequence according to the simple and basic scheme shown in Figure 8. The seed \( x_0 \) along with the control parameters is randomly selected. The generated bit-sequence is subjected to a statistical test to check for clear patterns and its suitability as a (PRNG).

2.3.1. Statistical Testing. The statistical analysis was conducted by the suite provided by the National Institute of Standards and Technology (NIST STS) [32]. NIST package comprises 15 tests that target the discovery of nonrandom patterns in the generated sequence. It tests for two types of hypotheses: null hypothesis \( H_0 \) (i.e., the sequence under test is random) and the alternative hypothesis \( H_a \) (i.e., the sequence under test is not random). After each test, a decision is produced with two possible outcomes, either accept \( H_0 \) or accept \( H_a \). These decisions are based on calculating the probability (\( P \)-value) and comparing it with a significance level (\( \alpha \)). For cryptographic applications, common values of \( \alpha \) are about 0.01. Hence, if \( P \)-value \( \geq \alpha \), then the null hypothesis is accepted (random sequence). If \( P \)-value \( < \alpha \), then the null hypothesis is rejected (not random).

After the test is conducted, the resultant \( P \)-values are counted and distributed over ten subintervals in the range of [0, 1]. Overall distribution of these \( P \)-values is determined using the incomplete gamma function

\[
P-values = \text{igamc}(9/2, (\chi^2/2)),
\]

where \( \chi^2 \) is the chi-square of the resultant \( P \)-values of a given test and is calculated using
Figure 5: Cross-correlation between two sequences $S_1$ and $S_2$ generated by (1): (a) same parameters, (b) with $p_{10} = p_{110} + d$, (c) $p_{210} = p_{211} + d$, (d) $L_{19} = L_{110} + d$, and (e) $L_{29} = L_{211} + d$, where $d = 10^6$. The results show that there is no correlation between the sequences and that the proposed map is very sensitive to any small change in any of the control parameters.

$$
\chi^2 = \frac{10}{m} \sum_{i=1}^{10} \frac{(F_i - (m/10))^2}{(m/10)},
$$

where $m$ is the sample size and $F_i$ is the number of $P$-values in the $i^{th}$ subinterval. NIST considers the sequences to be uniformly distributed if $P$-values $\geq 0.0001$.

The generator is tested by producing $m = 1000$ sequences with a length of $10^6$ bits each ($m = 100$ should be sufficient with respect to $\alpha = 0.01$; however, we choose $m = 1000$ to increase the accuracy of our test and prove the robustness of our system). For each sequence, $\{x_0, p_1, p_2, L_1, L_2\}$ were randomly chosen and the 15 tests were applied with the results listed in Table 1. To interpret these empirical results, an acceptable proportion range called the confidence region is determined using

$$
\text{confidence interval} = \hat{p} \pm 3 \sqrt{\frac{\hat{p}(1-\hat{p})}{m}},
$$

where $\hat{p} = 1 - \alpha$ and $\alpha = 0.01$. For any test, the evidence that the data is not random is manifested when a result proportion falls outside this interval.

Substituting in (6), we get the confidence interval of [0.9805607, 0.9994392]. The proportion of sequences that passes a test and the confidence region is shown in Figure 9, where all proportions lie within the confidence interval region. This is an indication that (1) can be used as PRBG suitable for cryptographic applications.

3. Application to Image Encryption

Throughout the years, various techniques were devised to achieve the highest possible levels of encryption efficiency [33]. Like other types of data that these methods have been applied to, image encryption is a real challenge due to the evolution in both cryptanalysis techniques and the electronic devices used to apply them [34]. In 1998, Fridrich et al. [35] introduced a scheme based on the baker map; since then, various schemes based on chaotic maps were given in literature [36–51]. The algorithms based on the tent map and skew tent maps have also been investigated. For example, Li et al. [52] proposed image encryption based on a pure chaotic tent map. However, [53] showed that this system suffered from multiple drawbacks since it is very simple and depends only on the diffusion step while omitting the permutation phase. The skew tent map was used in [54] with the orthogonal matrix and in [55] with cellular automata to produce a robust and secure image encryption schemes.

In this section, we introduce a new image encryption scheme based on the presented 1D chaotic function along with its security analysis.
3.1. The Proposed Scheme. The proposed scheme (Figure 10) has four rounds with each round consisting of five horizontal stages as follows:

1. The dimension $M \times N$ of the image ($I$) is calculated and fed along with a key to $X = f(k_1)$ and $Y = g(k_2)$. Each function will produce a stream of chaotic random-like numbers with the length $(l + M \times N \times 4 + G \times 3)$, where $l$ is the number of eliminated iterations from each chaotic function and $G$ is a gap length used to separate subsequences.

2. Four subsequences $X_1, X_2, X_3,$ and $X_4$ are extracted from $X$ transformed to $M \times N$ matrices.

3. Four permutation matrices $PM_1, PM_2, PM_3,$ and $PM_4$ are formed using $X_1, X_2, X_3,$ and $X_4$.

4. According to the round number, an image is permuted ($PI^n$) using the permutation matrix $PM^n$.

5. Four subsequences $Y_1, Y_2, Y_3,$ and $Y_4$ are extracted from $Y$ with $Y^n$ used for the diffusion step of $PI^n$ to produce $C^n$.

3.1.1. Secret Key Structure. Two unique keys ($k_1, k_2$) are fed to $(f, g)$ for generating a random-like stream. As seen from Figure 11, the whole key is divided into two subkeys, with each containing the necessary parameters for (1) to operate efficiently, namely, $x_0, p_1, p_2, L_1, L_2, h$. As described in the previous section, $h$ is the height of the map and is usually equal to $L_2$. However, it was found that choosing $h = L_2 - \delta$, where $\delta$ is smaller than $1 \times 10^{-3}$ will produce numerical sequences with better uniform distribution and random-like properties, thus leading to expanding the key space and increasing the security of the system.

Here, we select the first five variables to be 32 bits each and $h$ to be 10 bits long. As a result, each key will have a length of 170 bits with an overall of 340 bits for the whole key. The first five variables in each key are calculated based on fixed-point representation using the following formula:

$$\text{var} = \sum_{i=1}^{30} 2^{-i}B_{i+1}. \tag{7}$$
Figure 7: Histogram for randomly selected control parameters showing uniform distribution. For the shown figures the parameters were randomly generated as (a) $L_1 = 1, L_2 = 2, h = L_2 - \delta, p_1 = 0.25, p_2 = 0.8$; (b) $L_1 = 1.3, L_2 = 2.5, h = L_2 - \delta, p_1 = 0.75, p_2 = 1.4$; (c) $L_1 = 3, L_2 = 7, h = L_2 - \delta, p_1 = 1.5, p_2 = 6.8$; and (d) $L_1 = 2, L_2 = 10, h = L_2 - \delta, p_1 = 1.4, p_2 = 9.7$.

Figure 8: Simple scheme for generating a binary sequence from the iterated output of (1).

Algorithm 1: 3D Lyapunov exponent initialise $\delta_0, i, N, x_0$

1. for $p_1 \leftarrow 0$ to $L_1$ do
2. for $p_2 \leftarrow L_1$ to $L_2$ do
3. run the map for $N$ times
4. $xa_0 \leftarrow x(N)$
5. $xb_0 \leftarrow x(N) + \delta_0$
6. for $i \leftarrow 1$ to $N$ do
7. run the map with initial $xa_0$
8. run the map with initial $xb_0$
9. $\delta_1 \leftarrow |xa(i) - xb(i)|$
10. $\lambda = \ln(\delta_1/\delta_0)$
11. readjust the perturbed trajectory using:
12. $x_{bo} = x_{ao} + \delta_0(x_{bo} - x_{ao})/\delta_1$
13. end for
14. end for
15. end for
Figure 9: All proportions passing the test lie within the confidence interval.

Figure 10: The proposed encryption scheme using two PRNG incorporating the chaotic map described in (1).

Figure 11: Key structure for the proposed scheme.
However, δ is calculated using

$$\delta = \sum_{i=-14}^{23} 2^i,$$

(8)

where var represents any of the variables \(x_{o'}, p_1, p_2, L_1, L_2\), and \(h\). \(B_{i+1}\) represents the bit position from left to right, structured for two bits as the integer part and 30 bits for the fraction. This directly tackles the weak key property discussed in [56] especially when using floating-point representation to construct the key.

3.1.2. Permutation Stage. The redundancy characteristic in a digital image leads to a high correlation between its adjacent pixels. Hence, breaking this correlation by pixel permutation is an important step for securing a scheme against statistical attacks. In this manuscript, we used a simple and effective permutation algorithm to attain this task (Figure 12). We start by using the dimension of the original image \((M \times N)\) and the subkey \(k_1\) to generate the following sequence using \(f(x, k_1)\):

$$X = \{x_1, x_{i+1}, x_{i+2}, x_{i+3}, \ldots, x_{i+M \times N + 4 + G \times 3}\},$$

(9)
where \( l \) and \( G \) are the eliminated iterations and the gap between the extracted subsequences. The next step is to extract four equal subsequences from \( X \) according to the following formulae:

\[
\begin{align*}
X^1 &= X(S_1; S_2), \\
X^2 &= X(S_3; S_4), \\
X^3 &= X(S_5; S_6), \\
X^4 &= X(S_7; S_8),
\end{align*}
\]

where

\[
\begin{align*}
S_1 &= I, \\
S_2 &= S_1 + M \times N - 1, \\
S_3 &= S_2 + G, \\
S_4 &= S_3 + M \times N - 1, \\
S_5 &= S_4 + G, \\
S_6 &= S_5 + M \times N - 1, \\
S_7 &= S_6 + G, \\
S_8 &= S_7 + M \times N - 1.
\end{align*}
\]

The extracted \( X^1, X^2, X^3, \) and \( X^4 \) are converted to 2D matrices with dimensions of \( M \times N \). For the first round, the 2D matrix from \( X^1 \) will be sorted first by columns and then by rows, while taking into account the retention of the original location of each cell. The pixels of the original image are then permuted according to these matrices. The same steps are used to permute the ciphered image from each round.

3.1.3. Diffusion Stage. To increase the resistance against differential attacks, any small change in the original image should lead to nonuniform spreading across the ciphered image. After the permutation stage, the diffusion phase is designed to attain this goal by replacing the bit level value of each encrypted pixel with the previously encrypted pixel and the generated key stream \( Y^n \) according to the following formula:

\[
c_i = A_i \oplus E_i, \tag{12}
\]

where \( A_i \) is equivalent to \( I \) or \( C^n \) according to the encryption round. \( E_i \) is defined by

\[
E_i = \begin{cases} 
\lfloor (1 + Y^n_{i}) \times 10^{10} \mod 256 \rfloor, \\
\lfloor (c_{i-1} \oplus [Y^n_{i} \times 10^{10} \mod 256]) (1 + Y^n_{i}) \times 10^{10} \mod 256 \rfloor.
\end{cases} \tag{13}
\]

Similarly, by reversing the steps in Figure 12 and using (12), decryption of the encrypted image can be achieved.

3.2. Experimental Results and Security Analysis. An encryption scheme is said to be robust if it withstands its security against all types of known attacks like key brute-force attack, statistical attacks, and differential attack. In this section, the proposed scheme is subjected to various security analysis methods to verify its robustness. All calculations were performed using a 64-bit double-precision floating-point representation. The experimental analysis was done using 64-bit MATLAB (R2015a) running under Windows 10 installed on a Core i7 2.2 GHz machine with 8 GB of RAM.

3.2.1. Security of the Key Space. In the presented algorithm, the secret key consists of two subkeys \( (k_1, k_2) \), with an overall of 340 bits composing the whole key. Hence, \( 2^{340} \) or ~10\(^{102} \) attempts are required to brute-force the key. This verifies that the key space is large enough to withstand a brute-force attack.

3.2.2. Key Sensitivity Analysis. An encryption scheme is sensitive to its secret key if (a) a change of one bit in the key will produce a completely different ciphered image and (b) a change of one bit in the key will produce a completely different decrypted image. In Figure 5, we proved the sensitivity of the proposed chaotic map to any small change in the initial condition or control parameters. Similarly, the effect of changing the least significant bit in the key on the encrypted image is depicted in Figure 13, where Figure 13(a) is the original image. Then, we used the values listed in Table 2 to initially construct the key \( \{k_1, k_2\} \). The key is then used to encrypt the original image as shown in Figure 13(b) (we call the encrypted image \( C_1 \)). The key is then used to encrypt the original image as shown in Figure 13(b) (we call the encrypted image \( C_1 \)). The result of this encryption is another encryption \( (C_2) \) of the original image but by flipping the LSB in one of the control parameters in the original key. To verify if changing only one bit produces a completely different encrypted image, absolute pixel-by-pixel subtraction was performed between \( C_1 \) and \( C_2 \) \( |C_1 - C_2| \). The result of this subtraction is shown in Figure 13(d). The noisy figure is an indication that two different encrypted images were produced with a slight change in the original key.

To further elaborate on this, the previously described test was repeated for all parameters in \( k_1 \) and \( k_2 \). The mean square error (MSE) and peak signal-to-noise ratio (PSNR) were calculated according to (14) and (15) and listed in Table 3. The obtained numbers confirm the sensitivity of the scheme to all key parameters. Since \( k_1 \) is for permuting the pixels and \( k_2 \) is for changing the gray levels, the results in the table prove that the scheme has good confusion-diffusion properties:

\[
\text{MSE} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |E_1(i, j) - E_2(i, j)|^2}{M \times N}, \tag{14}
\]

\[
\text{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\text{MSE}}}, \tag{15}
\]

3.2.3. Histogram Analysis. Histogram shows the distribution of color intensities in an image. A good encryption scheme would produce a ciphered image with a uniformly
distribut ed histogram even for images with weak color intensity distribution.

In Figure 14, each histogram of an original image shows unique intensity distribution, while all those of the encryption result exhibit a uniform shape. This indicates that the proposed algorithm can resist statistical attacks.

3.2.4. Correlation Analysis. Since the high correlation of adjacent pixels in an image makes it vulnerable to statistical attacks, a robust encryption scheme should be able to break this correlation in the vertical (V), horizontal (H), and diagonal (D) directions. To calculate the correlation coefficient \( r_{xy} \), \( S \) random pairs of adjacent pixels are selected and substituted in the following equations:

\[
\begin{align*}
\text{cov}(x, y) & = \frac{1}{S} \sum_{i=1}^{S} (x_i - E(x))(y_i - E(y)), \\
D(x) & = \frac{1}{S} \sum_{i=1}^{S} (x_i - E(x))^2, \\
D(y) & = \frac{1}{S} \sum_{i=1}^{S} (y_i - E(y))^2, \\
E(x) & = \frac{1}{S} \sum_{i=1}^{S} x_i, \\
E(y) & = \frac{1}{S} \sum_{i=1}^{S} y_i, \\
r_{xy} & = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}}.
\end{align*}
\]

Figure 13: Key sensitivity of the proposed algorithm. The effect of changing one bit in the key on the encrypted images: (a) original image, (b) encrypted image \( C_1 \) using \( x_0 = 0 \times 9AC32BC4 \) in \( k_1 \) (corresponding to 2.418162290068759918212890625 in decimal form using (7)), (c) encrypted image \( C_2 \) using \( x_0 = 0 \times 9AC32BC4 \oplus 0 \times 000000001 \) for \( k_1 \), and (d) absolute pixel-by-pixel difference \( |C_1 - C_2| \) indicating two different encrypted images were produced.

Table 1: NIST STS result for 1000 sequences generated by randomly selected \([x_0, p_1, p_2, L_1, L_2]\).

| Test                  | P-values  | Passing ratio | Result |
|-----------------------|-----------|---------------|--------|
| Frequency             | 0.023545  | 0.992         | PASS   |
| Block frequency       | 0.101917  | 0.984         | PASS   |
| Cumulative sums       | 0.548314  | 0.993         | PASS   |
| Runs                  | 0.17377   | 0.988         | PASS   |
| Longest run           | 0.420827  | 0.99          | PASS   |
| Rank                  | 0.727851  | 0.991         | PASS   |
| FFT                   | 0.678686  | 0.992         | PASS   |
| Nonoverlapping template | 0.968863  | 0.993         | PASS   |
| Overlapping template  | 0.06126   | 0.983         | PASS   |
| Universal             | 0.15991   | 0.991         | PASS   |
| Approximate entropy   | 0.544254  | 0.989         | PASS   |
| Random excursions     | 0.815329  | 0.9947        | PASS   |
| Random excursions variant | 0.878107  | 0.9947        | PASS   |
| Serial                | 0.972382  | 0.991         | PASS   |
| Linear complexity     | 0.029598  | 0.989         | PASS   |

Table 2: Values used to test for key sensitivity in the proposed algorithm (these values can be converted to their decimal form using (7)).

| \( k_1 \) | \( x_0 \) | \( p_1 \) | \( p_2 \) | \( L_1 \) | \( L_2 \) | \( \delta \) |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 \times 9AC32BC4 | 0 \times 3AB3C17D | 0 \times 7CCD1CEB | 0 \times 73C14EB6 | 0 \times ABDFC4A1 | 0 \times 7B241FCA | (0010100100) |
| 0 \times 73C5A427 | 0 \times 205AC54F | 0 \times 6213B75A | 0 \times 4147B217 | 0 \times 7B241FCA | (1111010111) |
A visual display for $S = 3000$ pixels of such calculations is shown in Figure 15. It is visibly clear that pixel correlation is strong and compacted in the original image, while it is scattered and weak in the ciphered image. The result of subjecting different images to that test is listed in Table 4. All the numbers for the ciphered images suggest weak correlation and robustness against statistical attacks.

3.2.5. Information Entropy Analysis. In physics, entropy is a measure of "knowledge", or its opposite "uncertainty" about a certain system. Applying this definition to cryptography shows that the less information we can extract from the ciphered message, the more secure the ciphering algorithm is. In 1949, Shannon [58] mathematically linked this definition to information security. For image encryption, this equation is given by

$$H(m) = - \sum_{i=1}^{2^n} P(m_i) \log_2(P(m_i)), \quad (17)$$

where $n$ is the number of bits representing the color intensity and $P(m_i)$ is the probability of a color intensity $m_i$ in an image $I$. For a grayscale image, $n = 8$ bits with total gray levels of 256 ($0 \sim 255$); if all gray levels in an image have the same probability $P(m_i) = (1/256)$ which is equivalent to a uniform histogram, then the result of (17) would be 8. This indicates that, as the entropy value increases (very close to 8), less information can be extracted from it. In other words, histogram and entropy are both quantitative measures of "uncertainty": one with visual representation and the other with a numerical result. In Table 5, the result for the entropy analysis of several ciphered images is listed and confirms the uniform distribution of gray levels in them. Another
important factor is to calculate the local Shannon entropy (LSE) of nonoverlapping portions of the image using

\[ H_{k,T_B}(I) = \sum_{i=1}^{k} \frac{H(I_{R_i})}{k}, \quad (18) \]

where \( k \) is the number of blocks, \( T_B \) is the number of pixels in each block, and \( H(I_{R_i}) \) is the Shannon entropy of the block. Table 5 shows the comparison of LSE of the proposed scheme with other schemes using significant value \( \alpha = 0.05 \), \( k = 30 \), and \( T_B = 1936 \). This led to a critical interval of

**Figure 15:** Correlation analysis using 3000 adjacent pixels, (a) original image, (b) vertical correlation, (c) horizontal correlation, (d) diagonal correlation, (e) ciphered image, (f) vertical correlation, (g) horizontal correlation, and (h) diagonal correlation.

**Table 3:** MSE and PSNR of two encrypted images: \( C_1 \) by the key in Table 2 and \( C_2 \) using the same key but changing one parameter.

| Changed parameter | New value | MSE \( \times 10^3 \) | PSNR (dB) |
|-------------------|-----------|----------------------|----------|
| \( x_0 \)          | 0 × 9AC32BC4, 0 × 00000001 ⊕ 0 × 9AC32BC5 | 10.858 | 7.7730 |
| \( P_1 \)          | 0 × 3AB3C17D, 0 × 00000001 ⊕ 0 × 3AB3C17C | 10.964 | 7.7290 |
| \( P_2 \)          | 0 × 7CC1D1CEB, 0 × 00000001 ⊕ 0 × 7CC1D1CEA | 10.942 | 7.7396 |
| \( L_1 \)          | 0 × 73C14EB6, 0 × 00000001 ⊕ 0 × 73C14EB7 | 10.935 | 7.7425 |
| \( L_2 \)          | 0 × ABDFCA4A1, 0 × 00000001 ⊕ 0 × ABDFCA4A0 | 10.892 | 7.7594 |
| \( \delta \)        | (0010100100)₂ (0000000001)₂ ⊕ (0010101010)₂ | 10.974 | 7.7268 |
| \( x_0 \)          | 0 × 73C5A427, 0 × 00000001 ⊕ 0 × 73C5A426 | 10.961 | 7.7320 |
| \( P_1 \)          | 0 × 205AC54F, 0 × 00000001 ⊕ 0 × 205AC54E | 10.934 | 7.7429 |
| \( P_2 \)          | 0 × 6213B75A, 0 × 00000001 ⊕ 0 × 6213B75B | 10.935 | 7.7423 |
| \( L_1 \)          | 0 × 4147B217, 0 × 00000001 ⊕ 0 × 4147B216 | 10.930 | 7.7445 |
| \( L_2 \)          | 0 × 7B241FCB, 0 × 00000001 ⊕ 0 × 7B241FCB | 10.956 | 7.7342 |
| \( \delta \)        | (111010101)₁₂, (0000000000)₁₂ ⊕ (111010101)₂ | 10.878 | 7.7653 |

**Table 4:** Vertical, horizontal, and diagonal correlation of some images and their encrypted ones (source of the original images [57]).

| File name | V-correlation | Original image | H-correlation | D-correlation | V-correlation | Ciphered image | H-correlation | D-correlation |
|-----------|---------------|----------------|---------------|---------------|---------------|----------------|---------------|---------------|
| 5.1.14    | 0.88037       | 0.94713        | 0.84129       | -0.008326     | -0.061198     | 0.057352       |               |               |
| 5.2.08    | 0.95097       | 0.94137        | 0.90008       | -0.017024     | -0.014425     | 0.074288       |               |               |
| 5.2.09    | 0.88681       | 0.89632        | 0.82119       | -0.018392     | 0.009652      | 0.069188       |               |               |
| 5.2.10    | 0.93430       | 0.93994        | 0.87766       | -0.033006     | -0.041143     | 0.010029       |               |               |
| 5.3.01    | 0.98366       | 0.97222        | 0.96749       | 0.037097      | 0.047065      | 0.075035       |               |               |
| 5.3.02    | 0.89503       | 0.90269        | 0.85448       | -0.022656     | 0.017415      | -0.014040      |               |               |
| 7.1.01    | 0.92372       | 0.95824        | 0.89554       | 0.042068      | -0.002003     | 0.039041       |               |               |
### Table 5: Entropy analysis of sample grayscale images.

| File name   | Global entropy | LSE       |
|-------------|----------------|-----------|
| 5.1.14      |                |           |
| 5.2.08      |                |           |
| 5.2.09      |                |           |
| 5.2.10      |                |           |
| 5.3.01      |                |           |
| 5.3.02      |                |           |
| 7.1.01      |                |           |

### Table 6: Typical values of $\mathcal{N}_a^\alpha$, $\mathcal{U}_a^{\alpha^-}$, and $\mathcal{U}_a^{\alpha^+}$ for different image sizes using $\alpha = 0.05$.

| File size | NPCR (%) | $\mathcal{N}_a^\alpha$ | UACI       | $\mathcal{U}_a^{\alpha^-}$ | $\mathcal{U}_a^{\alpha^+}$ |
|-----------|----------|-------------------------|------------|---------------------------|---------------------------|
| 256 × 256 |          | 99.5693                |            | 33.2824%                  | 33.6447%                  |
| 512 × 512 |          | 99.5893                |            | 33.3730%                  | 33.5541%                  |
| 1024 × 1024 |       | 99.5994                |            | 33.4183%                  | 33.5088%                  |

### Table 7: NPCR and UACI score results for 26 images with different sizes.

| File name       | Score% | Status  | NPCR test result | Reference [59] | Reference [60] | Reference [61] | UACI test result | Reference [59] | Reference [60] | Reference [61] |
|-----------------|--------|---------|------------------|----------------|----------------|----------------|------------------|----------------|----------------|----------------|
| Dimension       |        |         |                  |                |                |                |                  |                |                |                |
| 256 × 256       |        |         |                  |                |                |                |                  |                |                |                |
| 5.1.09.tif      | 99.5941 | Pass    | 99.603           | 99.599         | 89.415        | 33.4722        | Pass             | 33.552         | 33.290         | 0.698          |
| 5.1.10.tif      | 99.5728 | Pass    | 99.636           | 99.627         | 16.201        | 33.6179        | Pass             | 33.453         | 33.273         | 0.064          |
| 5.1.11.tif      | 99.5743 | Pass    | 99.942           | 99.635         | 52.489        | 33.5225        | Pass             | 33.586         | 33.323         | 0.412          |
| 5.1.12.tif      | 99.5758 | Pass    | 99.792           | 99.617         | 28.144        | 33.3374        | Pass             | 33.453         | 33.535         | 0.440          |
| 5.1.13.tif      | 99.6459 | Pass    | 99.792           | 99.624         | 8.677         | 33.5497        | Pass             | 33.520         | 33.459         | 0.034          |
| 5.1.14.tif      | 99.6170 | Pass    | 99.621           | 99.632         | 48.378        | 33.5752        | Pass             | 33.440         | 33.396         | 12.094         |

| Dimension       |        |         |                  |                |                |                |                  |                |                |                |
| 512 × 512       |        |         |                  |                |                |                |                  |                |                |                |
| 5.2.08.tif      | 99.5918 | Pass    | 99.960           | 99.611         | 30.740        | 33.4202        | Pass             | 33.692         | 33.452         | 0.121          |
| 5.2.09.tif      | 99.6040 | Pass    | 99.876           | 99.632         | 23.050        | 33.3967        | Pass             | 33.548         | 33.444         | 0.090          |
| 5.2.10.tif      | 99.5987 | Pass    | 99.654           | 99.602         | 99.608        | 33.5028        | Pass             | 33.454         | 33.527         | 0.344          |
| 7.1.01.tif      | 99.6014 | Pass    | 99.957           | 99.613         | 11.129        | 33.4868        | Pass             | 33.648         | 33.525         | 0.044          |
| 7.1.02.tif      | 99.6056 | Pass    | 99.918           | 99.619         | 61.226        | 33.4162        | Pass             | 33.465         | 33.531         | 0.240          |
| 7.1.03.tif      | 99.3975 | Pass    | 99.849           | 99.610         | 10.535        | 33.5348        | Pass             | 33.273         | 33.401         | 0.041          |
| 7.1.04.tif      | 99.6006 | Pass    | 99.991           | 99.618         | 22.989        | 33.4449        | Pass             | 33.202         | 33.520         | 0.180          |
| 7.1.05.tif      | 99.6109 | Pass    | 99.942           | 99.586         | 94.707        | 33.4587        | Pass             | 33.830         | 33.505         | 2.971          |
| 7.1.06.tif      | 99.6113 | Pass    | 99.670           | 99.600         | 67.933        | 33.4813        | Pass             | 33.627         | 33.437         | 0.535          |
| 7.1.07.tif      | 99.5968 | Pass    | 99.983           | 99.608         | 47.197        | 33.4569        | Pass             | 33.609         | 33.511         | 0.185          |
| 7.1.08.tif      | 99.6117 | Pass    | 99.818           | 99.605         | 5.817         | 33.4746        | Pass             | 33.375         | 33.418         | 0.023          |
| 7.1.09.tif      | 99.6151 | Pass    | 99.874           | 99.617         | 58.374        | 33.4900        | Pass             | 33.530         | 33.405         | 0.229          |
| 7.1.10.tif      | 99.5941 | Pass    | 99.697           | 99.606         | 99.595        | 33.4539        | Pass             | 33.438         | 33.536         | 33.39          |
| boat12.tif      | 99.6009 | Pass    | 99.715           | 99.615         | 4.513         | 33.3759        | Pass             | 33.374         | 33.476         | 0.018          |
| gray21.tif      | 99.5917 | Pass    | 99.643           | 99.609         | 35.443        | 33.4828        | Pass             | 33.507         | 33.483         | 0.279          |
| ruler2.tif      | 99.6021 | Pass    | 99.637           | 99.623         | 8.248         | 33.4163        | Pass             | 33.415         | 33.353         | 0.032          |

| Dimension       |        |         |                  |                |                |                |                  |                |                |                |
| 512 × 512       |        |         |                  |                |                |                |                  |                |                |                |
| 5.3.01.tif      | 99.6078 | Pass    | 99.950           | 99.613         | 12.852        | 33.4706        | Pass             | 33.508         | 33.479         | 0.101          |
| 5.3.02.tif      | 99.6009 | Pass    | 99.982           | 99.601         | 39.601        | 33.4801        | Pass             | 33.514         | 33.482         | 33.419          |
| 7.2.01.tif      | 99.6010 | Pass    | 99.980           | 99.606         | 60.130        | 33.4664        | Pass             | 33.487         | 33.474         | 0.236          |
(7.901901305, 7.903037329). The table shows that the proposed scheme falls within these critical values for different image sizes.

3.2.6. Resist to Differential Attack Analysis. A robust encryption scheme should also be sensitive to small error in the plaintext. That is, while using the same key, a change in one pixel in the plain image would produce a completely different ciphered image. If a scheme possesses this property, then it has the ability to resist differential attacks.

\[
\text{NPCR}(C, C') = \sum_{i,j} \frac{D(i,j)}{T} \quad D(i,j) = \begin{cases} 
0, & \text{when } C(i,j) = C'(i,j), \\
1, & \text{when } C(i,j) \neq C'(i,j),
\end{cases}
\]

where \( T = M \times N \) is the number of pixels in the image. An ideal scheme would produce NPCR = 1 which implies that no relation between \( C \) and \( C' \) can be achieved. But since this value is difficult to attain, values very close to unity could be accepted under a certain criterion.

UACI measures the average intensity difference between \( C \) and \( C' \) using the following equation:

\[
\text{UACI}(C, C') = \frac{\sum_{i,j} |C(i,j) - C'(i,j)|}{T \times L}.
\]

where \( L \) is the maximum level of color intensity. Criteria for accepting the results of both tests were given in [41]. For NPCR, a test result is accepted when it is greater than or equal a one sided hypotheses \( \mathcal{N} \) with \( \alpha \) as its significance level:

\[
\mathcal{N}^* = \frac{L + \Phi^{-1}(\alpha) \sqrt{L/T}}{L + 1},
\]

where \( \Phi(\cdot)^{-1} \) is the inverse CDF of the standard normal distribution \( \mathcal{N}(0, 1) \).

For UACI test, the result of an encryption is considered a pass if its value falls inside the interval \([\mathcal{U}^*_-, \mathcal{U}^*_+]\) as follows:

\[
\begin{align*}
\mathcal{U}^- &= \mu_L - \Phi^{-1}(\frac{\alpha}{2}) \sigma_L, \\
\mathcal{U}^+ &= \mu_L + \Phi^{-1}(\frac{\alpha}{2}) \sigma_L
\end{align*}
\]

where \( \mu_L = \frac{L + 2}{3L + 3} \) and \( \sigma_L = \frac{(L + 2)(L^2 + 2L + 3)}{18(L + 1)^2 L} \).

In practice, number of pixels’ change rate (NPCR) and unified average changing intensity (UACI) are the two common methods for measuring this type of sensitivity. Consider two images \( I \) and \( I' \), where \( I' \) is slightly different from \( I \) by one pixel only. Encrypting both images with the same key will produce \( C \) and \( C' \).

NPCR measures the absolute number of pixels with the same position in \( C \) and \( C' \) that changed value.

| File name     | Size   | PSNR dB |
|---------------|--------|---------|
| 5.1.14.tif    | 256 × 256 | 28.43   |
| 5.2.08.tif    | 512 × 512 | 27.9893 |
| 5.2.09.tif    | 512 × 512 | 25.83   |
| 5.2.10.tif    | 512 × 512 | 27.9893 |
| 5.3.01.tif    | 1024 × 1024 | 29.13  |
| 5.3.02.tif    | 1024 × 1024 | 29.52  |

Table 8: Peak signal-to-noise ratio (PSNR) between the plain image \( P \) and the ciphered image \( C \).

| File name     | Size   | Encryption speed (second) |
|---------------|--------|---------------------------|
| 5.1.14.tif    | 256 × 256 | 0.3545                    |
| 5.1.09.tif    | 256 × 256 | 0.3548                    |
| 5.1.10.tif    | 256 × 256 | 0.3519                    |
| 5.2.08.tif    | 512 × 512 | 1.4598                    |
| 5.2.09.tif    | 512 × 512 | 1.4474                    |
| 5.3.01.tif    | 1024 × 1024 | 6.0516                    |
| 5.3.02.tif    | 1024 × 1024 | 5.9146                    |

Table 9: Speed analysis of the proposed scheme for different image sizes.

3.2.7. Perceptual Security: Peak Signal-to-Noise Ratio (PSNR). Another indication of encryption quality is to calculate the PSNR between the original image and the ciphered one.
where $M \times N$ is the image size and $P$ and $C$ are the plain and ciphered images, respectively. Table 8 lists the PSNR of multiple images, where values greater than 25 indicate good encryption quality.

### 4. Execution Speed Analysis

To test the execution speed for both the proposed map and encryption scheme, a laptop with Intel(R) Core(TM) i7-4702MQ CPU @ 2.20 GHz and 8 GB RAM was used to perform the experiment. The average execution speed of the proposed maps runs at 6.2 ms while the conventional tent map executes at an average of 5.2 ms. The running speeds of the proposed encryption scheme for different image sizes are listed in Table 9. These running speeds are acceptable considering the high level of security of the proposed scheme.

### 5. Conclusion

Recently, chaos theory has been linked to cryptographic applications. One-dimensional chaotic map has the simplest form of mathematical and hardware representation but suffers from many security problems like collapsing and limited effect of its control parameters. This paper introduced a 1D chaotic function with five variable control parameters in addition to the initial condition $x_0$. The function proved to be topologically mixing with dense periodic orbits and sensitivity to its initial condition and control parameters. A simple PRNG based on the map was constructed and tested using NIST test suites; the results gave a good indication of the strong cryptographic property of the presented map. Based on the presented map, a new image encryption scheme was presented. Several analyses proved the scheme to be robust against various types of attacks with good diffusion and confusion property.

### Data Availability

All data included in this study are available upon request by contact with the corresponding author.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah (Grant no. DF-222-144-1441). The authors, therefore, gratefully acknowledge DSR’s technical and financial support.

### References

[1] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer Security: Principles and Practice*, Pearson Education, London, UK, 2012.

[2] B. Furtl and D. Kiroskki, *Multimedia Security Handbook*, CRC Press, Boca Raton, FL, USA, 2004.

[3] A. Kerckhoffs, “La cryptographie militaire, ou, des chiffres usités en temps de guerre: Avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef. Librairie militaire de L. Baudoin,” 1883.

[4] P. Dushenski, “Kerckhoffs’ history and principles of military cryptography,” 2015, http://www.contravex.com/2015/03/04/kerckhoffs-history-and-principles-of-military-cryptography-translated-and-annotated/.

[5] P. L’ecuyer, “Tables of linear congruential generators of different sizes and good lattice structure,” *Mathematics of Computation of the American Mathematical Society*, vol. 68, no. 225, pp. 249–261, 1999.

[6] C. Fontaine, “Linear congruential generator,” *Encyclopedia of Cryptography and Security*, Springer, Berlin, Germany, 2011.

[7] J. Von Neumann, “13. various techniques used in connection with random digits,” *Applied Mathematical Sciences*, vol. 12, no. 36-38, p. 5, 1951.

[8] R. K. Bhullar, L. Pawar, and V. Kumar, “A novel prime numbers based hashing technique for minimizing collisions,” in *Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 522–527, Dehradun, Utrakhand, October 2016.

[9] G. Marsaglia, A. Zaman, and W. Wan Tsang, “Toward a universal random number generator,” *Statistics & Probability Letters*, vol. 9, no. 1, pp. 35–39, 1990.

[10] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, “Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers,” *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 223–233, 1995.

[11] P. Murali and G. Senthilkumar, “Modified version of playfair cipher using linear feedback shift register,” in *Proceedings of the 2009 International Conference on Information Management and Engineering*, pp. 488–490, Kuala Lumpur, Malaysia, April 2009.

[12] M. Alawida, A. Samsudin, J. S. Teh, and W. H. Alshoura, “Digital cosine chaotic map for cryptographic applications,” *IEEE Access*, vol. 7, pp. 150609–150622, 2019.

[13] S. S. Jamal, A. Anees, M. Ahmad, M. F. Khan, and I. Hussain, “Construction of cryptographic s-boxes based on mobius transformation and chaotic tent-sine system,” *IEEE Access*, vol. 7, pp. 173273–173285, 2019.

[14] Z. Hua and Y. Zhou, “Exponential chaotic model for generating robust chaos,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2019.

[15] C. Mira, “Chaotic dynamics: From the one-dimensional endomorphism to the two-dimensional diffeomorphism,” *World Scientific*, vol. 262, 1987.

[16] S. C. Phatak and S. S. Rao, “Logistic map: A possible random-number generator,” *Physical Review E*, vol. 51, no. 4, pp. 3670–3678, 1995.

[17] L. Shujun, M. Xuanqin, and C. Yuanlong, “Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography,” in *Proceedings of the International Conference On Cryptology in India*, pp. 316–329, Chennai, India, December 2001.

\[
\text{PSNR} = 10 \log_{10} \left( \frac{M \times N \times 255^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P(i, j) - C(i, j))^2} \right)
\]
[18] S. Li, X. Mou, Y. Cai, Z. Ji, and J. Zhang, "On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision," Computer Physics Communications, vol. 153, no. 1, pp. 52–58, 2003.

[19] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," International Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129–2151, 2006.

[20] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," International Journal of Bifurcation and Chaos, vol. 15, no. 10, pp. 3119–3151, 2005.

[21] M. Alawida, A. Samsudin, and J. S. Teh, "Enhancing unimodal digital chaotic maps through hybridisation," Nonlinear Dynamics, vol. 96, no. 1, pp. 601–613, 2019.

[22] M. Alawida, A. Samsudin, J. S. Teh, and W. H. Alshoura, "Deterministic chaotic finite-state automata," Nonlinear Dynamics, vol. 98, no. 3, pp. 2403–2421, 2019.

[23] Z. Hua, Y. Zhou, and B. Bao, "Two-dimensional sine chaotic oscillation system with hardware implementation," IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 887–897, 2019.

[24] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotic system for improving chaos complexity," IEEE Transactions on Signal Processing, vol. 68, pp. 1937–1949, 2020.

[25] I. Öztürk and R. Kılıç, “Cycle lengths and correlation properties of finite precision chaotic maps,” International Journal of Bifurcation and Chaos, vol. 24, no. 9, Article ID 1450107, 2014.

[26] W. S. Sayed, A. G. Radwan, A. A. Rezk, and H. A. Fahmy, “Finite precision logistic map between computational efficiency and accuracy with encryption applications,” Complexity, vol. 2017, pp. 1–21, Article ID 8692046, 2017.

[27] T. A. Elmanfaloty and E. Abou-Bakr, "Random property enhancement of a 1D chaotic PRNG with finite precision implementation." Chaos, Solitons & Fractals, vol. 118, pp. 134–144, 2019.

[28] L. Cong, W. Xiaofu, and S. Songgeng, "A general efficient method for chaotic signal estimation," IEEE Transactions on Signal Processing, vol. 47, no. 5, pp. 1424–1428, 1999.

[29] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, and V. Fernandez, "On the security of a new image encryption scheme based on chaotic map lattices," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 18, no. 3, Article ID 033112, 2008.

[30] R. L. Devaney, An introduction to Chaotic Dynamical Systems, Westview press, Boulder, CO, USA, 2008.

[31] J. C. Sprott, "Numerical calculation of largest lyapunov exponent," http://sprott.physics.wisc.edu/chaos/lyapexp.htm.

[32] L. E. Bassham, A. L. Rukhin, J. Soto et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.

[33] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, Berlin, Germany, 2010.

[34] A. Joux, "A tutorial on high performance computing applied to cryptanalysis," in Annual International Conference On the Theory and Applications Of Cryptographic Techniques, pp. 1–7, Cambridge, UK, April 2012.

[35] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," International Journal of Bifurcation and Chaos, vol. 8, no. 6, pp. 1259–1284, 1998.

[36] E. Solak, Ç. Çokal, O. T. Yildiz, and T. Bioykoğlu, "Cryptanalysis of Fridrich’s chaotic image encryption," International Journal of Bifurcation and Chaos, vol. 20, no. 5, pp. 1405–1413, 2010.

[37] J.-x. Chen, Z.-l. Zhu, C. Fu, and H. Yu, "A fast image encryption scheme with a novel pixel swapping-based confusion approach," Nonlinear Dynamics, vol. 77, no. 4, pp. 1191–1207, 2014.

[38] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3d chaotic cat maps,” Chaos, Solitons & Fractals, vol. 21, no. 3, pp. 749–761, 2004.

[39] Y. Wu, “Image encryption using the two-dimensional logistic chaotic map,” Journal of Electronic Imaging, vol. 21, no. 1, Article ID 013014, 2012.

[40] S. Somaraj and M. A. Hussain, “Performance and security analysis for image encryption using key image,” Indian Journal of Science and Technology, vol. 8, no. 35, 2015.

[41] Y. Wu, J. P. Noonan, and S. Agaian, “NPCR and UACI randomness tests for image encryption,” Journal of Selected Areas in Telecommunications (JSAT), vol. 1, no. 2, pp. 31–38, 2011.

[42] Z.-l. Zhu, W. Zhang, K.-w. Wong, and H. Yu, “A chaos-based symmetric image encryption scheme using a bit-level permutation,” Information Sciences, vol. 181, no. 6, pp. 1171–1186, 2011.

[43] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression-encryption scheme based on hyper-chaotic system and 2D compressive sensing," Optics & Laser Technology, vol. 82, pp. 121–133, 2016.

[44] Y. Zhou, L. Bao, and C. L. P. Chen, “A new 1D chaotic system for image encryption,” Signal Processing, vol. 97, pp. 172–182, 2014.

[45] Y.-Q. Zhang and X.-Y. Wang, "A new image encryption algorithm based on non-adjacent coupled map lattices," Applied Soft Computing, vol. 26, pp. 10–20, 2015.

[46] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," Information Sciences, vol. 273, pp. 329–351, 2014.

[47] L. Zhang, X. Liao, and X. Wang, "An image encryption approach based on chaotic maps," Chaos, Solitons & Fractals, vol. 24, no. 3, pp. 759–765, 2005.

[48] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," Optics and Lasers in Engineering, vol. 78, pp. 17–25, 2016.

[49] R. Roy, S. Mukund, G. J. Schut, D. M. Dunn, R Weiss, and M. W. W. Adams, "Purification and molecular characterization of the tungsten-containing formaldehyde ferredoxin oxidoreductase from the hyperthermophilic archaeon Pyrococcus furiosus: The third of a putative five-member tungstenzyme family," Journal of Bacteriology, vol. 181, no. 4, pp. 1171–1180, 1999.

[50] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, R. M. López-Gutiérrez, and O. R. Acosta Del Campo, "A RGB image encryption algorithm based on total plain image characteristics and chaos," Signal Processing, vol. 109, pp. 119–131, 2015.

[51] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using josephus problem and filtering diffusion," IEEE Access, vol. 7, pp. 8660–8674, 2019.

[52] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," Nonlinear Dynamics, vol. 87, no. 1, pp. 127–133, 2017.

[53] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," IEEE Access, vol. 5, pp. 6429–6436, 2017.

[54] J. Ahmad, M. A. Khan, F. Ahmed, and J. S. Khan, "A novel image encryption scheme based on orthogonal matrix, skew
tent map, and XOR operation,” *Neural Computing and Applications*, vol. 30, no. 12, pp. 3847–3857, 2018.

[55] B. Mondal, S. Singh, and P. Kumar, “A secure image encryption scheme based on cellular automata and chaotic skew tent map,” *Journal of Information Security and Applications*, vol. 45, pp. 117–130, 2019.

[56] J. S. Teh, M. Alawida, and Y. C. Sii, “Implementation and practical problems of chaos-based cryptography revisited,” *Journal of Information Security and Applications*, vol. 50, Article ID 102421, 2020.

[57] The USC-SIPI image database, http://sipi.usc.edu/database/.

[58] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[59] M. Alawida, J. S. Teh, A. Samsudin, and W. H. Alshoura, “An image encryption scheme based on hybridizing digital chaos and finite state machine,” *Signal Processing*, vol. 164, pp. 249–266, 2019.

[60] Z. Hua and Y. Zhou, “Image encryption using 2D logistic-adjusted-sine map,” *Information Sciences*, vol. 339, pp. 237–253, 2016.

[61] L. Huang, S. Cai, X. Xiong, and M. Xiao, “On symmetric color image encryption system with permutation-diffusion simultaneous operation,” *Optics and Lasers in Engineering*, vol. 115, pp. 7–20, 2019.