Journal of King Saud University – Computer and Information Sciences 35 (2023) 101595

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# A novel chaos-based permutation for image encryption

Moatsum Alawida

*Department of Computer Sciences and Information Technology, Abu Dhabi University, 59911 Abu Dhabi, United Arab Emirates*

ARTICLE INFO

*Article history:*
Received 13 March 2023
Revised 16 May 2023
Accepted 17 May 2023
Available online 23 May 2023

*Keywords:*
Chaos theory
Cryptography
Logistic chaotic map
Hash function
Image encryption

ABSTRACT

Image encryption is an essential method for ensuring confidentiality during transmission on open channels. Digital images have a high correlation with large redundant data, requiring alternative spatial handling during encryption. Digital chaos offers shared properties of randomness and sensitivity that can provide secure image encryption. The security and performance of chaotic image ciphers depend on the underlying digital chaotic map. Therefore, classical chaotic maps must improve security while maintaining implementation time. The main objective of this paper is to propose a new image encryption algorithm based on an enhanced chaotic map that offers high security and low time consumption. A new perturbed logistic chaotic map based on hybridizing backward and forward perturbation methods is introduced, which exhibits better chaotic features than other existing chaotic systems such as a large chaotic range, higher sensitivity, and randomness. Based on this, a new image encryption algorithm is proposed, employing a novel permutation operation and two substitution operations to achieve superior encryption speed and efficiency. The novel permutation operation based on a chaotic data sequence utilizes all chaotic states to generate 8-bit numbers. Each index of the resulting 8-bit number is compiled into one block. The plain image is then scanned in accordance with the randomized indices of each block. The two substitution operations involve an XORing operation for each pixel and a hashing process for each block. The newly proposed encryption algorithm can effectively encrypt images of varying sizes and types, transforming them into indecipherable cipher images that successfully pass through rigorous security tests, including differential attack analysis with a score of 28/28 and local Shannon entropy with a score of 25/28. The experimental findings showcase the superior performance of the proposed cipher over existing chaotic image ciphers, creating a cipher image that resembles noise and has the ability to resist a diverse range of cyber-attacks.



## 1. Introduction

Cryptography and chaos share many common features, which have motivated the development of various algorithms in a wide range of applications, including image ciphers (Hua and Zhou, 2016; Alawida et al., 2019a; Hua et al., 2018; Wang and Zhang, 2021; Gao et al., 2023a), hash functions (Alawida et al., 2021; Alawida et al., 2020), random number generators (Yu et al., 2019), authentication (Wang et al., 2022), and others (Alawida et al., 2023; Wafa' Hamdan et al., 2022). The crucial features are sensitivity and nonlinearity. Digital chaos is a nonlinear mathematical model that belongs to the class of dynamical systems. It is an iterative function and can be easily implemented on a computer. Depending on the number of chaotic states in the mathematical equation, it can be categorized into one dimension (1D), two dimensions (2D), and hyper dimensions (HD). When a 1D chaotic system is implemented on a computer, it is referred to as a chaotic map. The logistic map is a well-known example of a 1D chaotic map and is frequently used in logistic-based cryptography applications (Han, 2019; Alawida et al., 2022).

The use of the logistic map as a 1D chaotic system is limited by its simple chaotic behavior and lack of complexity in its chaotic state, resulting in security weaknesses such as low sensitivity to chaotic variables, limited chaotic range, and susceptibility to signal prediction methods (Fang et al., 2022; Hu et al., 2014). To address these issues and enhance the 1D chaotic maps, various methods have been proposed that fall into two categories: internal and external entropy sources (Cao et al., 2015). Internal methods use delayed chaotic states as the source of entropy to improve the chaotic map (Zheng et al., 2018), while external methods

*E-mail address*: moatsum.alawida@adu.ac.ae
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101595
1319-1578/

incorporate another chaotic map or a random function to improve the 1D chaotic map (Fan and Ding, 2019; Zheng et al., 2018; Cao et al., 2015). Although both methods have shown success in enhancing the chaotic system, the internal method is limited in range compared to the external method which provides better security but requires higher computational resources (Alawida et al., 2019b). Therefore, a simple structured external source can achieve a balance between security and performance.

Perturbation methods have been extensively employed to enhance the security of 1D chaotic maps (Hoang, 2022; Luo et al., 2021; Liu et al., 2021; Aziz et al., 2020). These methods involve perturbing either the chaotic state or control parameter using other values, and they consist of two components: mathematical functions and random numbers. The random numbers can be obtained from various sources, including chaotic states (Liu et al., 2017), random numbers (Liu et al., 2017), or other signals (Cao et al., 2015). Meanwhile, the mathematical functions can be modular or Xoring operations (Jun et al., 2018). Perturbation methods can be categorized into three types: forward, backward, and parameter perturbation (Alawida et al., 2019b). In the forward method, an entropy source is utilized to perturb the chaotic state after its calculation in the 1D chaotic map, which leads to an increase in the cycle length (Luo et al., 2021; Hua and Zhou, 2018). The backward method, on the other hand, employs an entropy source to perturb the chaotic state before its calculation, but it is still governed by the underlying 1D chaotic map. The parameter method uses an entropy source to perturb the control parameter to jump between the parameter range, but it is still governed by the range of the underlying 1D chaotic map. Therefore, the hybrid perturbation method is an optimal choice for achieving superior chaotic performance.

The utilization of digital chaos has played a significant role in designing many encryption algorithms for digital images in the area of image encryption, due to the shared properties between them (Fang et al., 2022). Chaotic maps such as 1D and HD maps are utilized to provide confusion and diffusion for image ciphers through the data sequence of chaotic points. In literature, several encryption algorithms using digital chaos have been proposed for the secure transmission of images (Aziz et al., 2020; Chai et al., 2017; Mansouri and Wang, 2020; Hua et al., 2018; Zhongyun Hua, 2018). They used permutation-substitution operations to achieve a better secure cipher. Chaotic points have been utilized in image encryption to permute the image pixels by changing their positions and removing correlation. Additionally, chaotic points have been employed to achieve diffusion by substituting the pixel values with other values and linking all the cipher pixels together. Hence, two operations hold significant importance in any image cipher. Chaos-based image encryption schemes commonly exhibit the following characteristics:

1. The proposed algorithms often require multiple scans of digital images, resulting in high computational complexity (Bezerra et al., 2021; Zhang et al., 2020).
2. Chaotic permutation operations typically employ sorting algorithms to generate new random indices for chaotic data sequences. However, this approach has high computational complexity (Alawida et al., 2019c; Talhaoui and Wang, 2021).
3. Classical chaotic maps such as the logistic map have limited chaotic ranges and low complexity behaviors, making them susceptible to dynamical degradation and signal prediction attacks. These issues could potentially compromise the security of the image cipher and make it easier to crack (Hua et al., 2018).
4. Deriving a secret key from a plain image to achieve high diffusion in a single round of image encryption is not in line with the principles of cryptography (Shannon, 1948).
5. The efficiency of a chaos-based image cipher is directly related to the computational complexity of the chaotic system used. Therefore, developing an effective image cipher based on chaos requires finding a balance between computational efficiency and security. This trade-off has been noted in prior research (Hua and Zhou, 2016; Fang et al., 2022; Alawida et al., 2019c).

This paper presents solutions to address various security issues in image encryption. The first solution involves improving the logistic map by using hybrid perturbation methods to generate a new perturbed logistic chaotic map with enhanced security and performance compared to existing ones. A simple structured model is employed to combine the backward and forward perturbation methods, which enhances the chaotic behavior of the logistic map to produce high-quality results. The new perturbed map preserves the computational complexity of the underlying unimodal logistic map and enhances chaotic features such as unpredictability, sensitivity to chaotic variables, enlarged chaotic range, ergodicity, and uniform distributions. Experimental results demonstrate that the new perturbed logistic map exhibits better security and chaotic quality. Utilizing the new perturbed map, a new chaos-based image encryption algorithm is introduced as the second solution to address the security vulnerabilities observed in existing image ciphers. The secret key is generated separately from the plaintext, ensuring independence and usability for multiple transmissions. The secret key is used to generate chaotic variables, which are then iterated through the new map to generate a chaotic data sequence. A novel permutation operation based on the new chaotic map is proposed, which is faster and requires only a single scan of the image. A substitution operation is then performed via two-round encryption. Experimental results and security analyses demonstrate that the proposed algorithm can encrypt plain images of any size and various types, creating a noise-like image with superior security compared to other chaotic image ciphers. It also resists different well-known attacks. This work introduces several new contributions, which can be summarized as follows.

1. A new perturbed logistic chaotic map has been introduced, which utilizes hybrid perturbation methods (backward and forward) to generate better chaotic performance while maintaining a reasonable time cost for implementation.
2. A new image encryption technique has been proposed, utilizing the new perturbed logistic map. To enhance the encryption, a novel permutation operation has been developed based on the high-quality characteristics of the new map.
3. An estimation of the chaotic performance of the new maps and other existing maps has been carried out, and the results demonstrate that the new map exhibits excellent chaotic performance when compared to the others.
4. Based on the simulation and security analysis performed, it has been demonstrated that the proposed image encryption algorithm is capable of encrypting different types of images into unreadable cipher images. Moreover, it has been found to provide a higher level of security when compared to several existing chaotic image encryption algorithms.

The remaining sections of this paper are structured as follows: Section 2 presents a literature review, while Section 3 provides an overview of the new perturbed logistic map and its chaotic analyses. Section 4 proposes the new image encryption algorithm. Section 5 presents experimental results and discusses their implications. Finally, Section 6 concludes the paper with some closing remarks.

## 2. Literature review

In literature, many works used two operations in different encryption rounds, as well as scanning the plain image many times to achieve the cipher target. Thus, scanning a plain image in different rounds and applying permutation operations leads to high-demand resources and costs in complexity computation (Bezerra et al., 2021; Zhang et al., 2020).

A new image encryption technique based on permutation-diffusion was proposed by Enayatifar et al. (2019). Their approach involved the utilization of DNA sequences and cellular automata to perform two critical operations. Firstly, the large image was divided into several sub-images, which were then transformed into 1D arrays. Secondly, a secure cipher was achieved through the use of index-based permutation. Talhaoui et al. have introduced a new method of image encryption based on a 1D cosine chaotic system (Wang et al., 2021). Their research shows that this new system exhibits improved chaotic behaviors and performance compared to previous methods. The image encryption was carried out using a parallel mode of the shifted scrambling diffusion method. However, despite its promising features, the proposed cipher employed a classical structure for scrambling and its security was found to be inadequate.

Mansouri and Wang enhanced the Arnold system for digital image encryption (Mansouri and Wang, 2021). They incorporated an improved Arnold system with a shuffle operation to achieve confusion and diffusion for the cipher images. The resulting cipher exhibited high sensitivity and resistance to well-known attacks. However, the proposed algorithm may not be resistant to other types of cryptanalysis attacks, and the security needs to be verified after multiple rounds of the Arnold system. A novel medical image encryption algorithm was proposed by Muthu and Murali, utilizing a newly developed 1D chaotic map (Muthu and Murali, 2022). This map provided a larger space for the secret key and generated the key stream at a faster rate. Diffusion and confusion were achieved through the use of a shuffle operation. However, the proposed cipher for medical images demonstrated weak diffusion performance.

Shahna and Mohamed introduced a new image encryption method based on chaos (Shahna and Mohamed, 2021). The key stream was generated using Lorenz's chaotic map, and pixel-level and bit-level scramble and diffusion operations were applied at two levels. While the proposed algorithm demonstrated a highly secure cipher image, the time consumption was prolonged due to the bit-level scrambling. A new image encryption algorithm based on the logistic map has been proposed by Wang and Gao (2020). The algorithm utilized a 2D logistic-adjusted sine chaotic map to generate the chaotic data sequence for encrypting the digital image. Two encryption rounds were achieved by combining the Boolean network with the chaotic data sequence. The proposed algorithm demonstrated improved security, but the use of the 2D chaotic map was computationally intensive.

In another study (Hua et al., 2019), Hua et al. (2019) introduced a novel technique to improve classical chaotic maps like the logistic, sine, and tent maps. Their approach involved using the cosine-transform method in combination with two seed maps, resulting in enhanced chaotic performance and the generation of several new chaotic maps. Furthermore, the authors proposed a new image encryption algorithm that utilized scrambling and substitution operations, with a total of four rounds of encryption. However, the time required to produce the cipher image was long. Li et al. (2017) have proposed a simple new algorithm for chaotic image encryption, utilizing the classical tent map. The keystream was produced from the chaotic data sequence, and the digital image was encrypted directly using the Exclusive OR (XOR) operation. Unfortunately, the effectiveness of this new chaotic image encryption algorithm was very weak and was subsequently breached by Wu et al. (2017) in their paper. In the same paper, Wu et al. developed an image encryption algorithm based on Li's cipher, which was subsequently also later broken by Zhu and Sun (2018).

Gao et al. (2023b) propose a 3D model encryption technique that utilizes a 2D chaotic system created through the combination of logistic map and infinite collapse (2D-LAIC), along with semi-tensor product (STP) theory. Compared to classical chaotic systems, 2D-LAIC demonstrates superior dynamical behavior. The study's comprehensive security and experimental analysis shows that the proposed algorithm, 3DME-SC, is highly efficient and effective.

## 3. Perturbed logistic chaotic map

This section introduces the perturbed logistic map, a new digital chaotic model based on the logistic map, and the perturbation method. The section also discusses its chaotic performance and complexity.

### 3.1. Logistic chaotic map

The logistic chaotic map is a 1D nonlinear dynamical system that can exhibit chaotic behaviors. It is widely used in various applications because of its properties, such as a simple structure that does not require many resources. The map has one chaotic state and one control parameter and can produce chaotic behaviors in a limited region of the control parameter range. Because of these properties, it is vulnerable to attacks and its control parameter can be easily predicted. Mathematically, the logistic map can be defined as follows:

$$F_L(s_{i+1}, r) = F_L(r \times s_i \times (1 - s_i)) \tag{1}$$

where $s_i$ is the chaotic state (point, system variable), for which $s_0$ is the initial condition bounded between 0 1nd 1, and $r$ is a control parameter with a range of $[0, 4]$. $i$ is an iteration index.

### 3.2. Perturbed logistic map

Perturbation methods have demonstrated their ability to improve classical digital chaotic maps, with the forward method proving to be the most effective. In this paper, a new chaotification method is proposed that combines the forward and backward perturbation methods to create a hybrid approach. This approach produces a better chaotic range than classical methods and extends the chaotic parameter range. To achieve this hybrid method, two carefully selected operations are added to the logistic map in order to maintain its simple structure. The first operation involves a mathematical equation given by

$$C_i = \frac{r}{s_i} \tag{2}$$

where $r$ is the control parameter and $s_i$ is the current chaotic state. This $C$ function is used as the input to the logistic map as the chaotic state, and the absolute function is used to ensure a positive output.

This method is known as the backward perturbation method, where the input is updated in each iteration using the $C$ function. The $C$ function always produces a large value since the chaotic state $s_i$ is a small value between 0 and 1. The logistic map is then rewritten using the following equation:

$$C_{i+1} = abs(r \times C_i \times (1 - C_i)) \tag{3}$$

Here, the absolute function is denoted by *abs*. On the other hand, the forward perturbation method is a straightforward mathematical model, represented as one operation under modular function. To

obtain the new chaotic state, two perturbation methods are applied to the logistic map. The forward method is expressed as:

$$s_{i+1} = (\frac{C_{i+1}}{C_i}) mod 1 \tag{4}$$

where $mod1$ is the modular function that removes all integer values and generates fraction values, thereby ensuring that all chaotic states remain between 0 and 1. The perturbed logistic map can be expressed in a different form as follows:

$$s_{i+1} = (\frac{abs(r \times \frac{r}{s_i} \times (1 - \frac{r}{s_i}))}{\frac{r}{s_i}}) mod 1 \tag{5}$$

This perturbed logistic map exhibits a larger chaotic parameter range and greater complexity of chaotic behavior compared to the original logistic map. The hybrid perturbation method offers excellent security to the underlying logistic map. The perturbed logistic map can be simplified to:

$$s_{i+1} = (abs(r \times (1 - \frac{r}{s_i}))) mod 1 \tag{6}$$

The new perturbed map and the logistic map have the same number of operations, yet the new perturbed map yields better results. The functions used in the hybrid perturbation method were carefully selected for the following reasons:

- Both the forward and backward functions involve the same mathematical operation, namely $\frac{a}{b}$.
- The denominator in this mathematical function has fewer values than the numerator, which increases the randomness in the chaotic points.
- The mathematical function consists of a control parameter and chaotic state without any external entropy source.
- The backward perturbation is used as a cascade system, while the forward perturbation is used as a combination method.
- The modular function is a simple function that removes integer values and retains fraction values as a result.

To demonstrate the chaotic behavior of the proposed chaotic map, the following mathematical proofs are presented.

**Lemma 1.** *Under the modulo operation, the perturbation function $\frac{r}{s_i}$ displays chaotic behavior whereas the logistic map only shows chaotic behavior for r values between 3.67 and 4. The proposed chaotic map is primarily controlled by the perturbation function $\frac{r}{s_i}$ under the modulo operation.*

**Proof**: Let begin with Eq. 6, which defines $s_{i+1}$ as $s_{i+1} = |r - \frac{r^2}{s_i}| mod 1$, where $s_i \in (0, 1)$ and $r \in (0, \infty)$. Let $|D|$ be the absolute difference between the two parts of Eq. 6, namely $r$ and $\frac{r^2}{s_i}$. If $|D|$ is a real number, which is implemented using the IEEE 754 double floating point number, then the value $|D|$ will be under the control of $s_i$. This can be expressed mathematically as:

$$\begin{aligned} &|r - \frac{r^2}{s_i}| = |D|, \quad Then\ |D| has\ two\ cases \\ &1:, |D| \approx \quad |r - r^2|, \qquad s_i < 0.5 \\ &2:, |D| \approx \quad |P|, \quad |P| > r^2,\ s_i \geqslant 0.5 \end{aligned} \tag{7}$$

Thus, the value of $|D|$ will oscillate between $|r - r^2|$ and $|\frac{r^2}{s_i}|$ (or $|P|$) based on the value of $s_i$. The modulo operation at the end of Eq. 7 simply removes the integer portion of the final result. □

Based on Alawida et al. (2022), it is possible to provide additional evidence for the chaotic nature of the proposed map by examining its sensitivity to initial conditions. Therefore, the following mathematical proof serves to reinforce the chaotic properties of the proposed chaotic map.

**Lemma 2.** *The fractional function $\frac{r}{s}$ is chaotic and highly sensitive to initial conditions whereas the logistic map only has chaotic behavior in the range of $r \in (3.67, 4)$. $\frac{r}{s}$ is chaotic under the modulo operation and has high sensitivity because a small change to $s_i$ will result in a large value (after solving for $\frac{r}{s}$) prior to applying the modulo operation.*

**Proof** Suppose $D$ is the difference between two values $s_0$ and $y_0$. If $D$ is a small real number (implemented using the IEEE 754 double floating-point number), then the first iteration of the function $\frac{r}{s_i}$ will result in a large value ($\gg r$), which can be expressed mathematically as follows:

$$\begin{aligned} &|s_0 - y_0| = |D| \\ &\left||r - \frac{r^2}{s_0}| - |r - \frac{r^2}{y_0}|\right| = |D'| \\ &|\frac{r^2}{y_0} - \frac{r^2}{s_0}| = |D'| \\ &|r^2 \times (\frac{1}{y_0} - \frac{1}{s_0})| = |D'| \end{aligned} \tag{8}$$

After the first iteration, we can observe that $|D'| > |D|$ because $|r^2 \times (\frac{1}{y_0} - \frac{1}{s_0})| > |s_0 - y_0|$. In subsequent iterations, the difference between $s_0$ and $y_0$ will grow exponentially, causing the two initial conditions to diverge over time. □

The amplification of sensitivity in the chaotic model is due to the large value of the inverse of the difference between two consecutive chaotic states, $\frac{r}{s_i}$, which is always a large value since the chaotic states $s_0$ and $y_0$ are real numbers between 0 and 1. This is because, after only one iteration, there is a large difference between two consecutive states, and the modulo operation removes the integer portion of the final result.

On another side, the Jacobian matrix, which represents the first derivative of a function $f\prime$, to Eq. 6, the resultant expression is $\frac{r^2}{s_i^2}$. Since $s_i$ is a fractional value, it is decreased under the quadratic exponent. Consequently, when $r > 1$, the output of the Jacobian matrix is always a large and positive value. Fig. 1 depicts the phase state of the perturbed logistic map after applying the Jacobian matrix. 50 control parameters are used ranging from 1 to 50 and 5000 chaotic states between 0 and 1. The results show that all values of $\frac{r^2}{s_i^2}$ are positive and large.

Finally, $\frac{r}{s_i}$ under modulo operation also exhibits chaotic behavior, as small perturbations in $r$ or $s_i$ can lead to large changes in the output. Therefore, $\frac{r}{s_i}$ is a chaotic function, while both $\frac{r}{s_i}$ and the logistic map are chaotic under the modulo operation.

### 3.3. Chaotic performance

The improved logistic map is based on two perturbation methods, which enhance its chaotic behavior and complexity compared to the classical logistic map. This subsection presents various evaluations to demonstrate the strengths of the perturbed map. The evaluations include a bifurcation diagram showing the chaotic behavior of the control parameter, sensitivity, and chaoticity tests using the Lyapunov exponent, fuzzy correlation dimension testing for fractal dimension and attractor's strangeness, evaluation of complexity and dynamical degradation using state-mapping network (SMN), respectively. In addition to the perturbed map and classical logistic map, other recent new chaotic maps were also used in these evaluations.

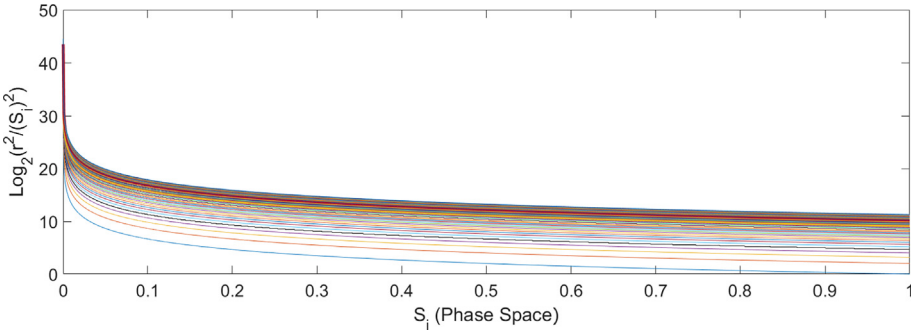



**Fig. 1.** Jacobian matrix of the perturbed logistic map, $r \in (0, 50)$ and $s_i \in (0, 1)$.

### 3.3.1. Bifurcation diagram

The bifurcation diagram is an essential tool to visualize chaotic behavior, where the shaded area in the diagram represents the chaotic behavior generated by a chaotic system with respect to a control parameter. It serves as a useful indicator for the keyspace in chaos-based encryption, where an increased shaded area along the range of control parameters leads to a larger keyspace, making it harder to attack using brute force methods. A comparison between the new map and other maps, including the classical logistic map and others (Han, 2019; Wang and Zhang, 2021), is shown in Fig. 2. For a fair comparison, the range of control parameters has been kept the same for three maps. The perturbed logistic map demonstrates a more fully shaded area than the others along $r \in (0, 4]$, while the logistic map has a smaller range with many periodic regions that disappear in the chaotic range of the perturbed logistic map. Another Fig. 3 illustrates the perturbed logistic map along the control parameter $r \in (0, 100]$ to demonstrate its large chaotic behavior with no unshaded area. The perturbed logistic map generates chaotic behavior in the non-chaotic area of the logistic map, which helps increase the keyspace for encryption algorithms that rely on it. In comparison with others, the perturbed logistic map shows more chaotic behavior.

### 3.3.2. LE

Chaotic behaviors are characterized by unpredictability and sensitivity. Unpredictability refers to the difficulty of predicting new chaotic points based on previous ones or finding patterns in chaotic point sequences over time. Sensitivity, on the other hand, measures how two chaotic data sequences diverge from each other over time, starting from very close initial conditions and with the same control parameter. To quantify these features, the Lyapunov exponent (LE) is a numerical measure of the rate of divergence between two chaotic data sequences with infinitesimally close initial conditions and the same control parameter. Positive LE values indicate highly sensitive and chaotic behavior, while zero or negative values indicate non-chaotic behavior, where the data sequence remains in a small region in phase space and cannot move freely to other regions.

The LE values of the perturbed logistic map and other maps are presented in Fig. 4, which shows that the perturbed map has positive LE values over the control parameter range $r \in (0, 4]$, while other maps have some negative LE values in the same range. This suggests that the perturbed map exhibits more chaotic behavior, unpredictability, and sensitivity compared to other maps, making it a suitable choice for chaos-based encryption algorithms that require diffusion and confusion features. Fig. 5 displays the perturbed map with LE values over the larger control parameter range of $r \in (0, 100]$. The perturbed map has positive LE values over this interval, indicating that it can also be used for chaos-based encryption since a larger chaotic range provides a larger keyspace.

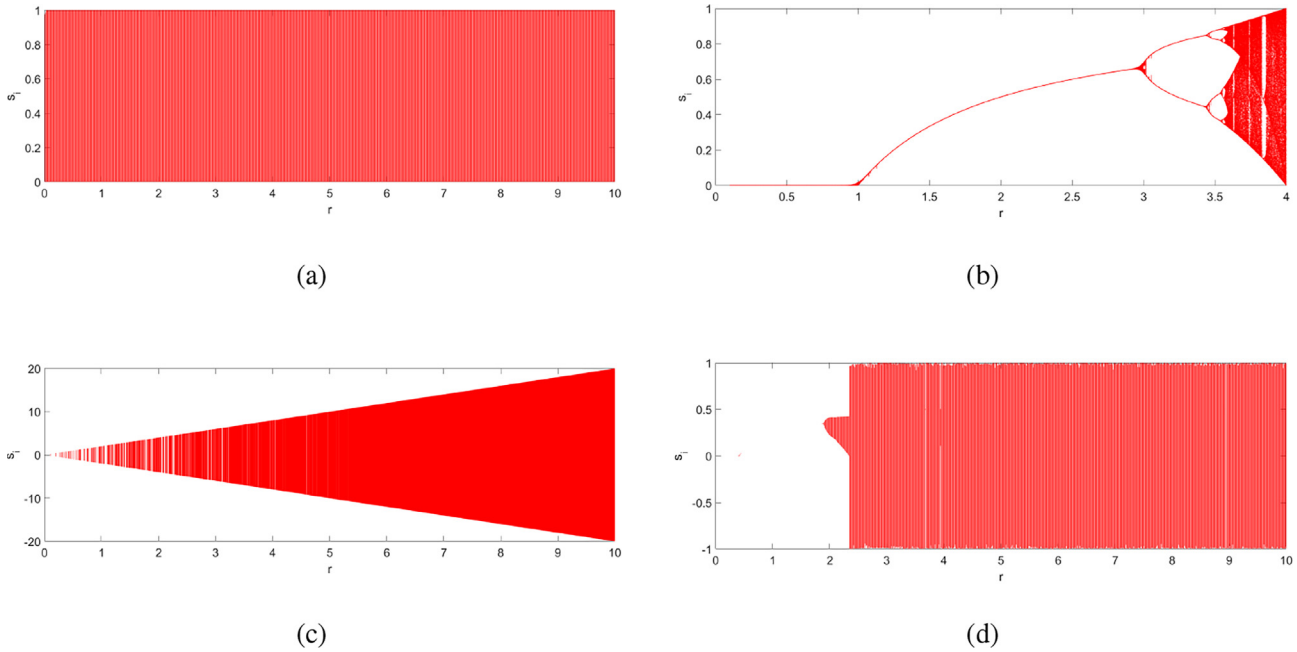



**Fig. 2.** Bifurcation diagrams of (a) perturbed logistic map, (b) logistic map, (c) modified logistic map (Han, 2019), (d) and C1DNSM map (Wang and Zhang, 2021).

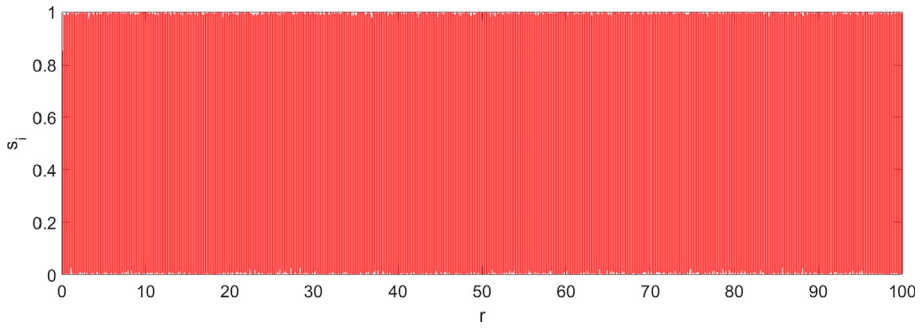



**Fig. 3.** Bifurcation diagrams of perturbed logistic map, $r \in (0, 100)$.

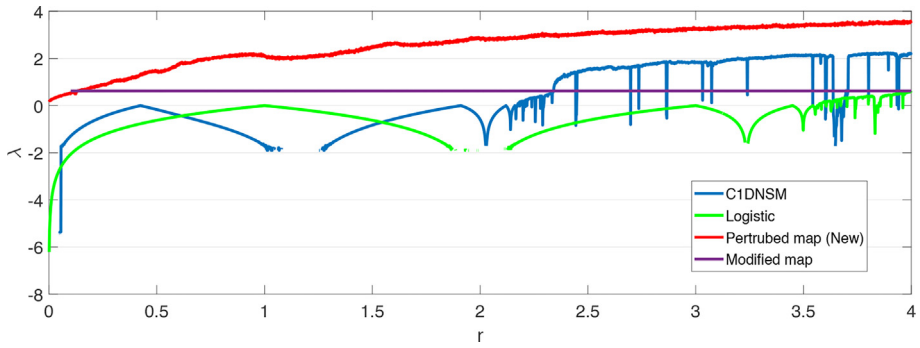



**Fig. 4.** LE values for the perturbed logistic map and others.

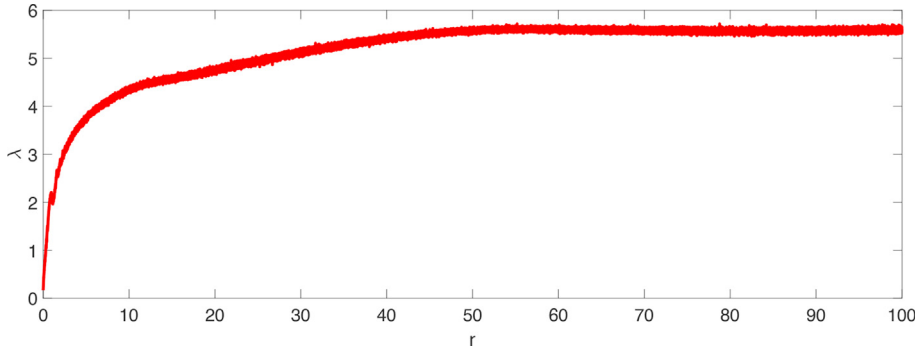



**Fig. 5.** LE values of the perturbed logistic map $r \in (0, 100]$.

*3.3.3. Fuzzy correlation dimension*

A phenomenon occurs in chaotic behavior where chaotic points flood and stretch irregularly across different locations in phase space as time progresses toward infinity. This phenomenon is known as a strange attractor, which arises in nonlinear systems, particularly chaotic ones. The strange attractor exhibits maximal chaotic and nonlinear properties. One of the estimators used to determine the strangeness of a chaotic system is the correlation dimension (CD), which can predict the geometries of the strange attractor in phase space (Grassbergert and Procaccia, 1983). The CD is computed using three components: the embedding dimension, correlational integral, and the Heaviside function. The fuzzy correlation dimension (FCD) is another estimator that provides more accurate results than the CD. The FCD employs a fuzzy membership function instead of the Heaviside function and includes the distance between chaotic points under the correlation integral. The estimated FCD can be found in Grassbergert and Procaccia (1983).

Higher FCD values indicate more complex and irregular behavior, high fractal dimension, and strange attractors in chaotic points. The FCD was calculated for the perturbed map and other maps along the interval of the control parameter. Fig. 6 displays the FCD values for all maps, with the perturbed map having larger values indicating high nonlinear and strangeness features. High dimensionality values indicate high nonlinearity in chaos-based encryption.

*3.4. Dynamical degradation analysis*

The difference between a real analog chaotic system and a digital chaotic map is significant. Analog systems use electronic circuitry and oscillations to simulate chaotic behavior, while digital chaotic maps are implemented using limited devices such as computers. Chaotic data sequences generated by analog systems are non-repetitive and have an infinite number of chaotic states, making them challenging to use in digital systems due to the time required for conversion and modulation.

In contrast, digital chaotic maps have finite precision and limited chaotic states, resulting in repeated chaotic states after a

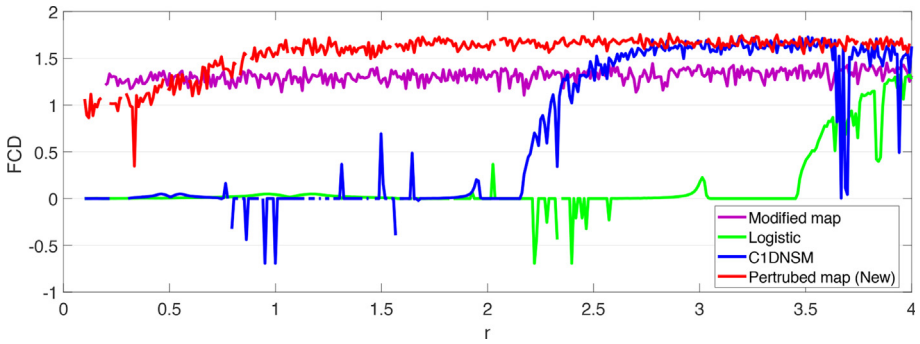



**Fig. 6.** FCD values for the perturbed logistic map and others.

certain time. Chaotic data sequences generated by digital maps can take two forms: fixed points or cycles. Fixed points repeat a single chaotic state indefinitely, while cycles repeat a set of chaotic states at regular intervals. Longer cycle lengths are preferred in digital chaos to shorter ones. However, digital chaos with short cycles is known as dynamical degradation, which results in undesirable features such as low complexity, low ergodicity, strong correlation, and non-uniform distribution.

To investigate the effect of dynamical degradation on chaotic maps, various methods can be employed to measure the number and length of cycles. One such method is SMN. For this analysis, fixed-point numbers with 6-bit precision are used. 64 initial states are labeled from 0 to 63, and the control parameter $R = 3.9999$ is used for both the perturbed and logistic maps. Each initial state is iterated once to generate the next state, which is one of the 64 states. A link is drawn between the initial state and the new chaotic state, and the links between all states are then combined to form a network. The network can be used to study the length and number of cycles or fixed points.

The comparison of SMN values for both maps is shown in Fig. 7. The perturbed map demonstrates a longer cycle length in two cycles, while the logistic map has only one cycle and a fixed point. This suggests that the perturbation method improves the complexity and prolongs the cycle of the underlying logistic map with minimal precision. The analysis is estimated for cycle lengths, starting from bit precision 4 up to 24. The cycle length for each chaotic map at each bit of precision is being calculated. Fig. 8 shows the cycle length for four chaotic maps, and it is observed that the perturbed chaotic map has a longer cycle as the bit precision increases.

It should be noted that this analysis uses 6-bit precision, and a higher precision such as 64-bit would result in even more complex and nonlinear behavior.

## 4. Image encryption algorithm

This section presents a new and innovative image encryption algorithm that combines a hashed block with a new chaotic map. The algorithm, illustrated in Fig. 9, comprises three phases: key generation, block generation, and permutation-substitution operations. In the key generation phase, a perturbed logistic chaotic map generates a chaotic data sequence, using floating-point numbers converted from a secret key as initial conditions and control parameters. Dynamic-sized blocks are generated based on the chaotic sequence, with block size depending on the secret key. The encryption process employs one permutation round and two substitution rounds to achieve a high-quality cipher. Each block is then hashed using the QUARK hash function, and the resulting hash value is used to encrypt the next block, resulting in a highly secure cipher image with non-correlated pixels. The proposed algorithm is fast and efficient compared to other existing methods. The following subsections provide a detailed description of the proposed algorithm.

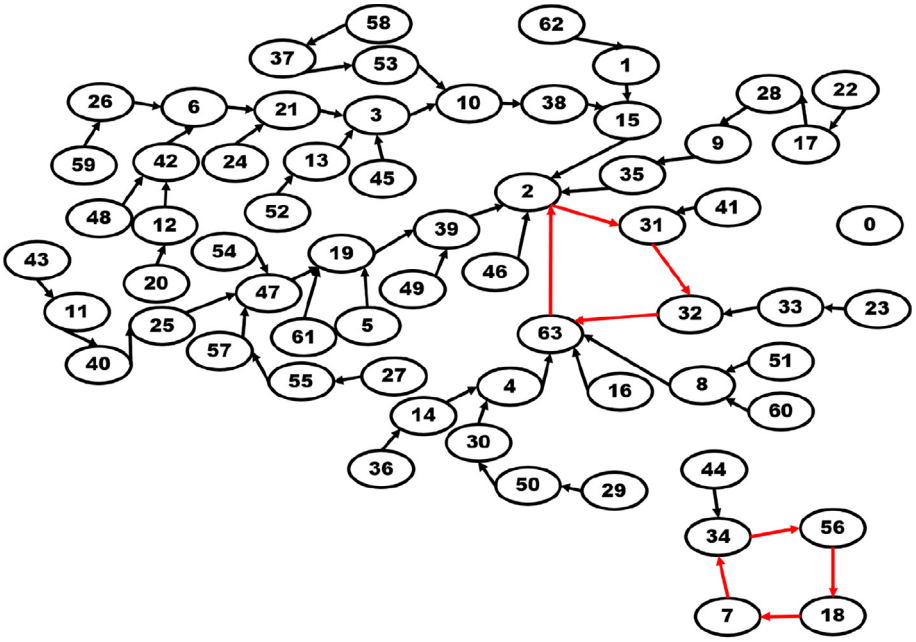
### 4.1. Key generation phase

In the realm of chaos-based image encryption algorithms, the secret key is often derived from initial conditions and control parameters. Some algorithms utilize multiple chaotic maps or even a combination of three-dimensional and hyperchaotic maps in order to increase the secret key space. The secret key can be represented as a floating-point number, which is convenient for chaotic encryption purposes. Alternatively, some algorithms use a bitstream as the secret key, which is then converted to floating-point numbers via a quantization function. Both methods have their pros and cons, which are compared in Table 1. This paper presents a new approach to using multiple initial conditions for a 1D chaotic map, combined with the use of quantization for the secret key.

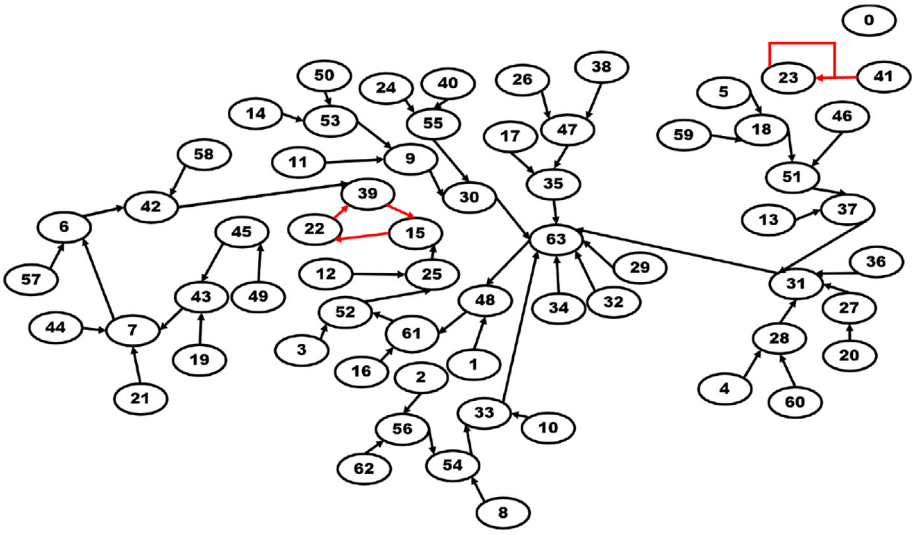
In symmetric-key ciphers, both the sender and receiver use the same secret key, eliminating the need for a third party to manage it. The secret key is shared between the parties via a secure channel, allowing for multiple images to be encrypted and decrypted during multiple transmissions. The secret key should adhere to the requirements of symmetric-key ciphers, having a key space larger than 128-bits. Going beyond this number does not significantly enhance security, but rather affects the efficiency of the algorithm. A brute force attack, which involves guessing the secret key by trying all possible options in the key space, is still unable to break 128-bits encryption. Therefore, this proposed algorithm uses 128 bits as the secret key.

The perturbed logistic map uses one initial condition $x_0$ and one control parameter $r$. The range of the initial condition is between 0 and 1, and it needs to be represented as a floating number in 52-bit significand format (IEEE-754, double precision). According to the IEEE-754 double-precision representation, it uses 1 sign bit, an 11-bit exponent, and a 52-bit significand, where the 11-bit exponent does not count for the initial condition. The control parameter has two values, the exponent and the significand. In this algorithm, the exponent is not considered for the control parameter and it can be any number and is not included in the secret key. Therefore, with 52-bits for the control parameter and 52-bits for the initial condition, the total becomes 104-bits. Another initial condition must be used to reach the 128-bit requirement for the secret key.

The proposed algorithm employs fixed-point representation for real numbers. Here, $U(j,k)$ denotes that $j$ and $k$ bits are used to represent the integer and fractional portions of a real number, respectively. To transform a $U(0,k)$ fixed-point number into its respective floating-point value, the following equation is used:




(a)




(b)

**Fig. 7.** SMN values of (a) perturbed logistic map, (b) logistic map.

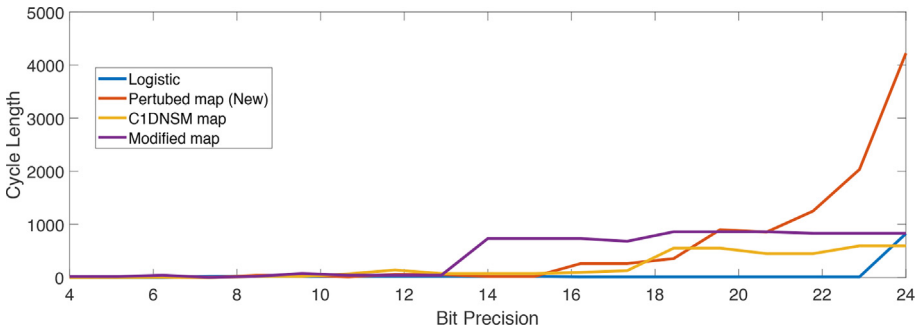



**Fig. 8.** Cycle analysis for the perturbed logistic map and others, ($s_0 = 0.11, r = 3.99$).

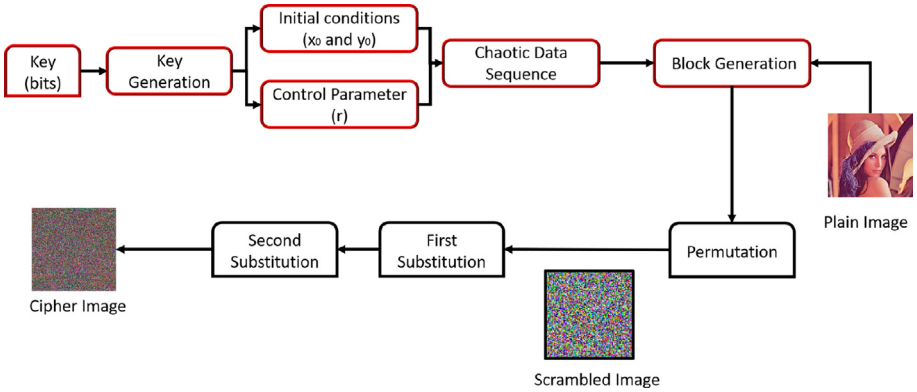



**Fig. 9.** Framework of the proposed image encryption algorithm.

**Table 1**
Comparison between quantization and floating-point methods.

| | Quantization | Floating-Points |
|---|---|---|
| Pre-processing | Requires conversion of bit stream to chaotic variables | No conversion needed |
| Flexible key size | Requires a mechanism to adjust key size | Increase number of chaotic maps |
| Calculation | Easy to calculate | Requires rounding process |
| Sensitivity | All bits are linked and more sensitive | Depends on number of chaotic maps and variables |

$$NR(x) = \sum_{L=1}^{k} \frac{bit_i}{2^L} \tag{9}$$

where $bit_i$ is the value of bit $L$ in the fixed-point representation of $x$. The steps to generate two initial conditions and a control parameter are as follows:

1. The control parameter $r$ is calculated using the bits from position 1 to 52 as

$$r = \sum_{L=1}^{52} \frac{bit_i}{2^L} \tag{10}$$

2. The first initial condition $x_0$ is calculated using the bits from position 50 to 100, and then compute this equation

$$x_0 = \frac{\sum_{L=50}^{100} \frac{bit_i}{2^L} + r}{2} \tag{11}$$

3. The second initial condition $y_0$ is calculated using the bits from position 76 to 128, and then compute this equation

$$y_0 = \frac{\sum_{L=76}^{128} \frac{bit_i}{2^L} + x_0}{2} \tag{12}$$

4. The perturbed logistic map is used to generate the chaotic data sequence using $r$ and $x_0$ for half the size of the plain image, and then $y_0$ is used to generate the second half.
5. The $y_0$ is perturbed using the last chaotic point $x_{\frac{3MN}{2}}$ of the perturbed logistic map as $y_0 = \frac{y_0 + x_{\frac{3MN}{2}}}{2}$.

A minor alteration in the secret key results in a new chaotic data sequence, and the dissimilarity between the two chaotic data sequences is substantial due to the high LE values and quick exponential divergence within a few iterations in the perturbed logistic map.

### 4.2. Block generation

In this section, a novel method for generating blocks using the perturbed logistic chaotic map is proposed. The secret key is used to generate chaotic variables, which consist of two initial conditions and a control parameter. The chaotic data sequence is generated based on the size of the plain image, using the example of a color plain image with a size of $M \times N \times 3$. This image has three channels (red, green, and blue), where the number of rows and columns are represented by $M$ and $N$, respectively. As a result, a chaotic data sequence of $3MN$ floating point numbers between 1 and 0 is generated. These chaotic points are then converted into 8-bit numbers as below,

$$Data_i = mod(x_i \times 2^{26}, 256) \tag{13}$$

where $Data_i$ refers to the 8-bit format representation of chaotic points, and $2^{26}$ is used to enhance the randomness of the data sequence. This value is the midpoint of the 52-bit floating-point format. The index $i$ of the chaotic point in the generated data sequence ranges from 1 to $3MN$.

The 8-bit generated data sequence, $Data_i$, ranges from 0 to 255 and has a highly ergodic and uniform distribution, ensuring that all values are generated with roughly equal frequency and making it difficult to predict the next value in the sequence. The perturbed logistic map demonstrates these secure chaotic characteristics, which result in diffusion and confusion in image cipher and resistance against well-known attacks, such as plaintext known attacks and histogram attacks (Noshadian et al., 2020), as shown in Section 3.3.

The process of creating non-overlapping blocks involves extracting the index of each value in the data sequence and assigning it to its respective block. There are 256 blocks in total according to the 8-bit format, each named based on its value, with block zero being the first one, block one being the second, and so on until block 255. When a new value is generated by the perturbed logistic map, its index is immediately placed in the block that corresponds to its value. For example, if the data sequence contains a value of 0, its index is added to $Block_0$, and if the data sequence contains a

value of 70, its index is added to $Block_{70}$, and so on. An illustration of this process can be seen in Fig. 10, which shows some blocks and their respective indices.

The search process for the non-overlapping blocks is straightforward and involves checking the chaotic point after converting it to an 8-bit format and then adding its index to the end of the corresponding block. A formal explanation of this process can be found in the Algorithm 1.

**Algorithm 1.** Block Generation

**Algorithm 1:** Block Generation

**Data:** Data, 3MN
**Result:** Block
$Block(0:255) = []$;
**for** *i=1 to* $3MN$ **do**
  **if** $i == \frac{3MN}{2}$ **then**
    $x_i = ((x_i + y_i) * 2^{26}) \ mod \ 1$;
  $x_i = abs(r * (1 - r/x_{i-1})) \ mod \ 1)$;
  $Data_i = (x_i \times 2^{26} \ mod \ 256$;
  $Block(data_i, end + 1) = i$;

Finally, the generation of 256 blocks is achieved through the use of indices of the chaotic points, due to the security provided by the perturbed logistic map. Each block contains more random and low-correlated values of the indices. Fig. 11 demonstrates the nearly equal frequency of values in the 256 blocks, with a slight difference in their sizes, leading to dynamic block sizes. This phase ends by merging all blocks into a 1D array, starting with the first block and followed by the second, and so on. The outcome of this phase is a 1D array that contains randomly generated indices of the chaotic points and has the same size as the original plain image.

### 4.3. Hash function

Quark is a lightweight hash function that was specifically proposed for Internet of Things (IoT) sensors, designed to handle a small amount of data (Aumasson et al., 2013). It is based on a sponge function and is a combination of the stream cipher Grain and the block cipher KATAN. The Quark family includes three instances: U-Quark, D-Quark, and S-Quark. The function features three types of permutation operations that are based on one linear feedback shift register (LFSR) and two nonlinear feedback shift registers (NFSR), as illustrated in Fig. 12. U-Quark, with a digest value of 136, has a rate of 8 and a capacity of 128. Security analysis has shown that U-Quark provides 64-bit security on 1397 digital gates, with a power consumption of 2.44 μWatts. In this paper, U-Quark is selected to generate the hash value for each block of the digital image, with the average value for each encrypted block being calculated by U-Quark and the first 64 bits of the hash value being extracted.

### 4.4. Image encryption phase

The blocks are generated by the power of the perturbed logistic and secret key, The index values in the blocks are highly random. To start the encryption phase of the image, the proposed algorithm focuses on providing two basic operations for image cipher; permutation and substitution. The index values for each block are used to provide a permutation of pixels and 8-bit chaotic points $Data_i$ themselves are used to provide substitution. The permutation is one round and substitution operations are achieved in two rounds to provide high secure image and achieve the diffusion feature for symmetric encryption. The permutation and substitution operations can be performed through the following steps:

**Index**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Block 0 | 369 | 1068 | 1283 | 1347 | 2047 | 2241 | 2295 | 2847 | 3553 | 4158 |
| Block 1 | 279 | 813 | 897 | 1484 | 2471 | 3791 | 3817 | 4054 | 4763 | 4944 |
| Block 2 | 225 | 242 | 334 | 913 | 1004 | 1225 | 1648 | 2049 | 2226 | 2559 |
| Block 3 | 129 | 187 | 642 | 1108 | 1125 | 1188 | 1217 | 1419 | 1445 | 1774 |
| Block 4 | 430 | 469 | 584 | 690 | 758 | 788 | 859 | 1874 | 2030 | 2114 |
| Block 5 | 849 | 1142 | 1339 | 1350 | 1426 | 2098 | 2474 | 2524 | 2706 | 2821 |
| Block 6 | 94 | 241 | 269 | 640 | 1222 | 1456 | 1643 | 1792 | 1869 | 2123 |
| Block 7 | 568 | 981 | 1250 | 1381 | 1477 | 1651 | 1921 | 2304 | 2792 | 3084 |
| Block 8 | 74 | 79 | 145 | 745 | 1139 | 1249 | 1259 | 1285 | 1460 | 2405 |
| Block 9 | 78 | 133 | 208 | 210 | 328 | 610 | 689 | 715 | 950 | 1509 |
| Block 10 | 148 | 566 | 1064 | 1147 | 1214 | 1244 | 1371 | 1443 | 1465 | 1475 |
| Block 11 | 166 | 293 | 1105 | 1610 | 3137 | 3261 | 3316 | 3917 | 4153 | 4175 |
| Block 12 | 193 | 399 | 685 | 1203 | 2189 | 2426 | 2632 | 2667 | 2740 | 2818 |

**Fig. 10.** Snippets of blocks and their respective indices.

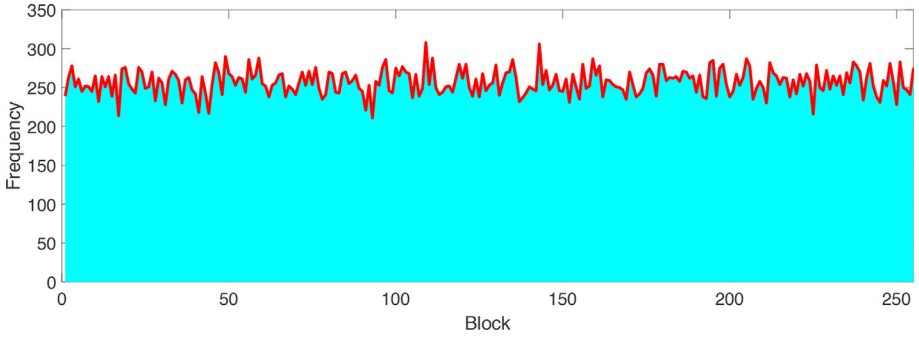



**Fig. 11.** Frequency of values for 256 blocks generated by the perturbed logistic map ($r = 7.3, x_0 = 0.3$ and $y_0 = 0.4$).

1. A plain image $P$ is transformed into a 1D array by reading the pixels row by row for all three channels.
2. The pixels of $P$ are read in sequence, starting with $Block_0$, then $Block_1$, and finally $Block_{255}$.
3. The reading process of the pixels is based on the index value in each block.
4. Each pixel of the plain image is encrypted using the corresponding value in the 8-bit chaotic points $Data_i$ and the encrypted value of the preceding pixel $C_{i-1}$.
5. After encrypting an entire block, the average value $AVB_k$ for the encrypted pixels is determined and a hash value $H_k$ for that average value is calculated using the QUARK hash function.
6. The output of the 64-bit hash function is converted into 8 values in 8-bit format $H_k(h_1, h2, \ldots, h_8)$, which are then Xored with the eight first pixels in the next block.
7. When the first round of encryption is completed for all the blocks, the second round begins by using the hash value of the final block of the first round to encrypt the first block again, then repeating the encryption process for the rest the blocks.

To clarify the permutation and substitution operations, Algorithm 2 includes all the encryption steps, including the finer details that were not previously mentioned.

**Algorithm 2.** Permutation and substitution algorithm

**Algorithm 2:** Permutation and substitution algorithm

**Data:** Plain Image P, 3MN, Block, Data
**Result:** Cipher Image C

*Convert P into a 1D array named C1;*
*Convert Block into a 1D array named Block1;*
*Re − read the pixels in C1 using the values from Block1;*
$H_k(1:8) = 0;$
**for** *Round =1 to 2* **do**
  $Pre = 0; k = 0; L = 0; j = 1;;$
  **for** *i=1 to 3MN* **do**
    **if** $Round == 1$ **then**
      $C1_i = P(Block1_i);$
    **if** $j <= 8$ **then**
      $C1_i = bitxor(H_{kj}, C1_i);$
      $j = j + 1;$
    $My = bitxor(Pre, uint8(Data_i));$
    $C1_i = bitxor(C1_i, My);$
    $k = k + 1;$
    $Pre - Block_k = C1_i;$
    **if** $k == length(Block_L)$ **then**
      $L = L + 1;$
      *Calculate the average (AVB) for the previous block;*
      $H_k = U - Quark(C1_{Pre-Block});$
      $k = 0;$
      $j = 1;$
$C = C1;$

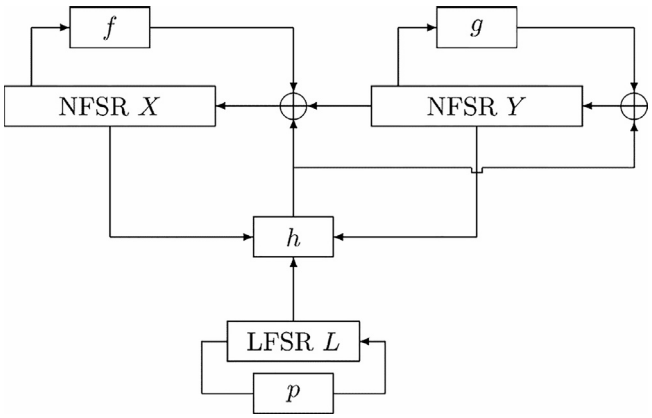



**Fig. 12.** Diagram OF the permutation operations of Quark.

Before testing the image cipher with various statistical attacks, there are some questions that need to be answered regarding the proposed algorithm. One of these questions is the reason for using a hashed block. The hashed block is utilized to attain diffusion across blocks, while the Xoring of each block's pixels with the preceding pixel provides diffusion within the block. As a result, the proposed algorithm provides double diffusion, both within and between the blocks.

The second question in the proposed algorithm concerns the use of the average value of a block. Its utilization is based on its simplicity and straightforward approach. By multiplying the average value by a significant factor, even minor changes within the block are amplified, leading to a greater effect on the final result of the hash value. This means that a small alteration in one pixel of the block will result in a new, distinct hash value, which will in turn impact subsequent blocks.

The use of two rounds of substitution operations in the proposed algorithm enhances the security of the image cipher. The first round spreads the data from the first block to the last but does not produce the reverse effect from the last block to the first. By employing two rounds, the entire image cipher is treated as a single, large block, providing high security and making it difficult to leak any pattern of information.

### 4.5. Decryption algorithm

Using the same encryption algorithm and secret key, the decryption process can reverse the encryption to recover the plaintext. In this study, decryption begins with the last block in the cipher image and calculates the hash value of the previous block to decrypt. Decryption then proceeds pixel by pixel, starting from the last pixel in the block and moving towards the first pixel in the same block. The first substitution operation in the decryption is performed, followed by the second substitution operation, and finally, the last permutation operation is applied to generate the plain image. Fig. 13 displays three types of plain images and their corresponding cipher images produced by the proposed algorithm. Furthermore, the algorithm successfully recovers the cipher images without any loss of pixels.

## 5. Experimental results and performance

In this section, a series of experiments were conducted on images including RGB color, gray level, and binary images. The tests were performed using Matlab R2022a on a machine equipped with Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz to 2.30 GHz Processor, 32 GB RAM, and running Windows 10 operating system. The SC-SIPI 'Miscellaneous' image dataset was used for the experiments (The usc-sipi image database), which contains 28 sample images of various sizes, some being $256 \times 256$ and others $1024 \times 1024$. This dataset has been utilized in several previous studies to assess the security level of proposed image encryption algorithms (Hua and Zhou, 2016; Alawida et al., 2019a; Hua et al., 2018).

In this study, a fair comparison with other chaotic image ciphers was made by selecting three highly cited and recent algorithms. These algorithms were re-implemented on the same machine. The results of these comparisons were recorded and reported. Other chaotic image ciphers were also selected for comparison, and their results were taken from their published papers.

### 5.1. Histogram analysis

The distribution of pixels in each image is unique. As previously mentioned, 256 gray levels are related to the number of colors in a single channel. The frequency of each gray level is calculated to determine the histogram, which is a visual representation based on the size of the image. A *flat* histogram indicates a uniform distribution of gray levels and eliminates any chance of pattern leakage or significant biases. The histogram comparison is presented in Fig. 14, which includes two plain images and their corresponding cipher images. The proposed algorithms result in a flat histogram for each channel in the images, regardless of the image content. This is because the substitution operations distribute the pixels evenly throughout the range of 0 to 255. However, the visual representation of the histogram is only an approximation and does not provide a precise measurement of pixel distribution. To better understand the uniformity of the cipher image, additional numerical measurements of the histogram are performed, including maximum deviation, irregular deviation, and deviation from the uniform histogram (DUH).

**Maximum deviation** is a metric employed to assess the variance in gray-level frequency between the original and encrypted images. This measurement is used to judge the quality of an image cipher. A higher maximum deviation suggests a greater difference between the histograms of the plain and encrypted images, as well as between their pixel values. This difference can be calculated by using the following equation.

$$M_D = \frac{D_0 + D_{L-1}}{2} + \sum_{i=1}^{L-1} D_i \tag{14}$$

where $D_i = |D_{C_i} - D_{P_i}|$ represents the absolute difference in the gray levels between the cipher image and the plain image at index $i$, and $L$ is the number of gray levels in the range of $[0, 256]$.

**Irregular deviation** is a metric that compares the differences in histograms to a uniform statistical distribution and is an extension of the deviation measure. The calculation of Irregular deviation can be represented as follows:

$$I_D = \sum_{i=0}^{L-1} |D_i - \gamma_H| \tag{15}$$

where $\gamma_H$ is the mean of the histogram values. A low value of Irregular deviation ($I_D$) signifies that the pixels in the cipher image are distributed evenly.

**DUH** measures the difference between the histogram of a cipher image and an ideal uniform distribution. The ideal uniform histogram can be computed as

$$H_C = \begin{cases} \frac{M \times N}{256}, & 0 \leqslant i \leqslant 255 \\ 0, & elsewhere, \end{cases} \tag{16}$$

where $M$ and $N$ are the width and height of an image respectively. The deviation of the cipher image's histogram from the ideal

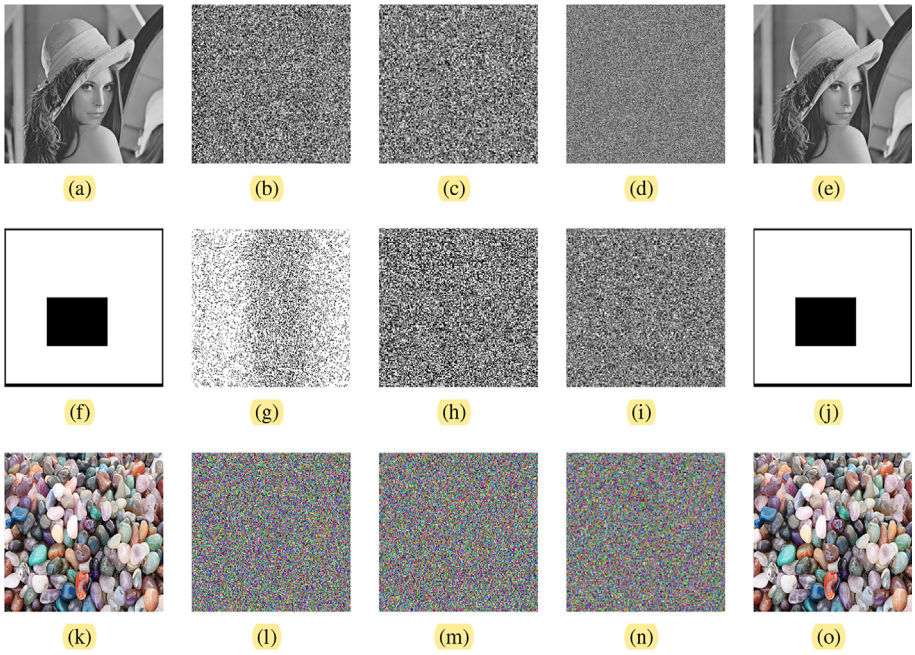



**Fig. 13.** Plain images and cipher images (a) plain image (Lena) size $256 \times 256$, (b) permutation, (c) substitution 1, (d) substitution 2, (e) decrypted image, (f) plain image (Small black box) size $128 \times 128$, (g) permutation, (h) substitution 1, (i) substitution 2, (g) decrypted image, (h) Plain image (Color) size $333 \times 500$, (i) permutation, (i) substitution 1, (j) substitution 2, (k) and decrypted image.

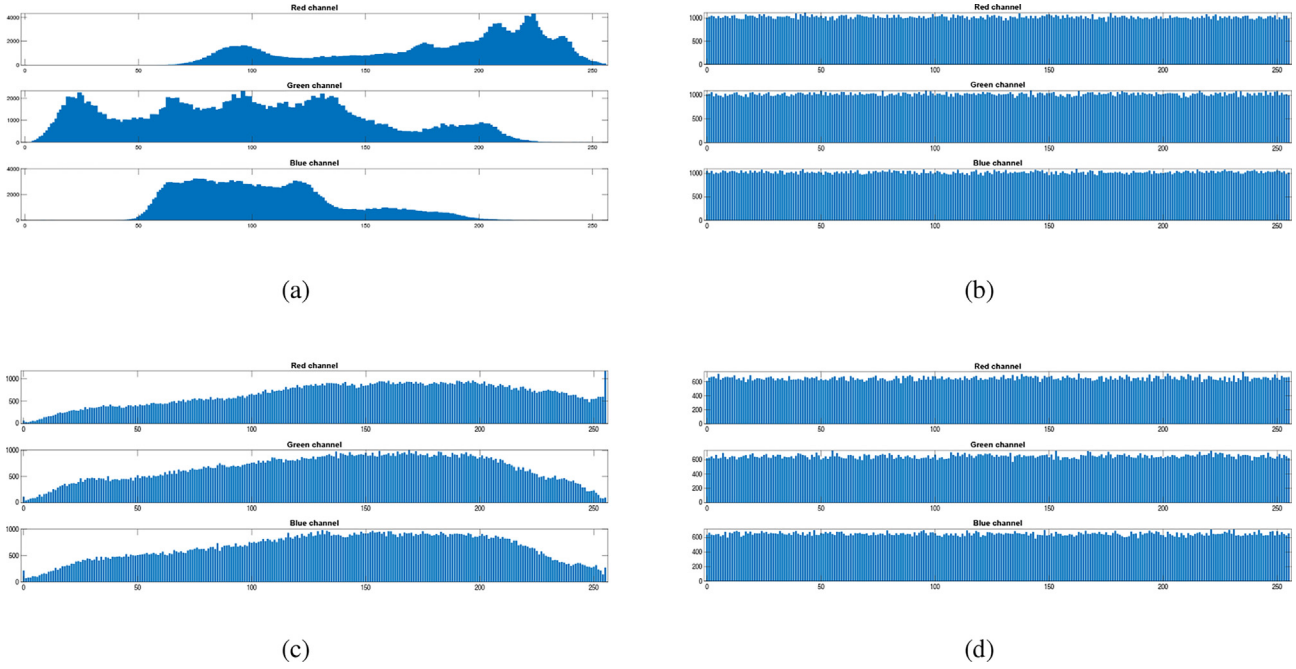



**Fig. 14.** Histograms of the plain and corresponding cipher images (a) Plain image (Lena), (b) cipher image (Lena), (c) Plain image (Color), and (d) cipher image (Color).

uniform histogram can be calculated by considering the dimensions of the image as

$$D_H = \frac{\sum_{C_i=0}^{255} |H_{C_i} - H_C|}{M \times N} \tag{17}$$

where $H_{C_i}$ is the gray value of the cipher image's histogram at index $i$. A low $D_H$ value suggests a low deviation and a uniform distribution of pixels in the cipher image.

The results of three histogram measurements were obtained and listed in Table 2. The proposed algorithm shows the uniform distribution and better resistance against histogram-based attacks compared to the other three existing algorithms, as indicated by its near-identical results and superior performance.

### 5.2. Correlation analysis

Plain images are a strong correlation between adjacent pixels in different directions. To break this correlation, the permutation

operation is a better solution. Thus, the weak correlation comes from the strong permutation. In this algorithm, a novel permutation approach based on the perturbed logistic map is proposed. The map is responsible to provide better permutation to the whole image cipher. As demonstrated in Section 3.3, the proposed map has high random and ergodic.

The correlation coefficients (CC) are determined by evaluating the relationships between pixels in three directions (vertical, horizontal, and diagonal). These coefficients can be calculated between the pixels within a single image and between the plain image and the encrypted image. It can be calculated as

$$C_r = \frac{COV(X,Y)}{\sqrt{D(X)D(Y)}}, \tag{18}$$

where $COV(x,y) = E([x - E(x)][y - E(y)]), E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $D(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2$. $x$ and $y$ are the values of adjacent pixels in the plain image or cipher image.

The optimal value for the lack of correlation is zero. Values close to zero in both positive and negative directions imply a minimal correlation between adjacent pixels in the encrypted image.

The results of the CC for the proposed algorithm on four images are presented in Table 3. The CC was computed for 4096 randomly selected pairs of adjacent pixels in three directions: diagonal, vertical, and horizontal. This calculation was done for both the plain image and the corresponding pixels in the encrypted image. The distributions of the CC are depicted in Fig. 15.

Additionally, Table 4 provides a comparison of the average CC results of the proposed algorithm with other algorithms across multiple images. The results demonstrate that the proposed algorithm is effective in reducing the correlation between adjacent pixels and outperforms the other algorithms.

### 5.3. Differential attack analysis

The measurement of this statistic is crucial in image encryption, as it assesses the discrepancy between two image ciphers obtained from two plain images with a single-bit difference in pixels. Consequently, any image cipher must be tested under a differential attack using the number of pixel change rate (NPCR) and the unified average change intensity (UACI) as the evaluation criteria. A slight variation between the two plain images leads to a marked difference between the cipher images generated using the same secret key.

The proposed algorithm boasts excellent diffusion properties, achieved through the following three factors:

- Two rounds of substitution operations, with the second round starting from the last block of the first round and encrypting the first block of the second round.
- The use of the hash value of each block to encrypt the subsequent blocks over two rounds.
- The previous encrypted pixel is utilized to encrypt each subsequent pixel within each block.

As a result, the proposed algorithm demonstrates high resistance to differential attack, as demonstrated in subsequent experiments. To compute the NPCR and UACI, two plain images with a single-bit difference are encrypted using the same secret key, yielding two cipher images $C_1$ and $C_2$. NPCR and UACI can then be estimated accordingly.

$$NPCR(C_1, C_2) = \frac{\sum_{i,j}^{M,N} D(i,j)}{MN} \times 100\% \tag{19}$$

and

$$UACI(C_1, C_2) = \frac{1}{MN}\sum_{i,j}^{M,N} \frac{|C_1(i,j) - C_2(i,j)|}{L} \times 100\% \tag{20}$$

respectively, where $MN$ represents the total number of pixels in a plain image, $L = 255$ is the maximum gray level value for an 8-bit pixel, and the function $D(i,j)$ is defined as:

$$D(i,j) = \begin{cases} 1, & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0, & \text{if } C_1(i,j) = C_2(i,j), \end{cases} \tag{21}$$

The results of these tests are not reliable, as noted in Wu et al. (2011). The tests should be evaluated based on their critical values, which are threshold values for different image sizes. One such critical value for NPCR is denoted as $N^*_\alpha$ and can be calculated as:

$$N^*_\alpha = \frac{L - \Phi^{-1}(\alpha)\sqrt{L/MN}}{L+1} \tag{22}$$

where $\alpha$ is the significance level, and $\Phi^{-1}$ is the inverse cumulative density function (CDF) of the normal distribution $N(0,1)$. To pass the NPCR test, an image cipher must have an NPCR value larger than $N^*_\alpha$.

**Table 2**
Histogram analyses comparison.

| Image | Proposed algorithm | | | Ref (Hua and Zhou, 2016) | | | Ref (Alawida et al., 2019a) | | | Ref (Hua et al., 2018) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M_D | I_D | DUH | M_D | I_D | DUH | M_D | I_D | DUH | M_D | I_D | DUH |
| 5.1.09 | 77132 | 39344 | 0.0476 | 76125 | 40125 | 0.0564 | 78125 | 40125 | 0.0488 | 78121 | 39400 | 0.0489 |
| 5.1.10 | 47802 | 26795 | 0.0538 | 46012 | 26912 | 0.0588 | 45856 | 27895 | 0.0598 | 45189 | 27983 | 0.0652 |
| 5.1.11 | 77114 | 43545 | 0.0526 | 75110 | 44251 | 0.0545 | 76125 | 44012 | 0.0552 | 77101 | 44542 | 0.0592 |
| 5.1.12 | 61629 | 47082 | 0.0547 | 60142 | 49512 | 0.0556 | 64138 | 48121 | 0.0569 | 62945 | 47092 | 0.0582 |
| Average | **65919** | **39183** | **0.0522** | 64347 | 40200 | 0.0563 | 66061 | 40038 | **0.0552** | 65839 | 46504 | 0.0579 |

**Table 3**
Correlation coefficient the first four images in dataset.

| Image | Plain image | | | Cipher image | | |
|---|---|---|---|---|---|---|
| | D | V | H | D | V | H |
| 5.1.09 | 0.9337 | 0.9338 | 0.8965 | 0.0003 | 0.0006 | −0.007 |
| 5.1.10 | 0.8429 | 0.9185 | 0.8713 | 0.0004 | −0.0017 | −0.021 |
| 5.1.11 | 0.7811 | 0.8358 | 0.8713 | −0.0009 | −0.0003 | 0.0144 |
| 5.1.12 | 0.9592 | 0.9375 | 0.9824 | 0.0001 | 0.00012 | 0.0174 |

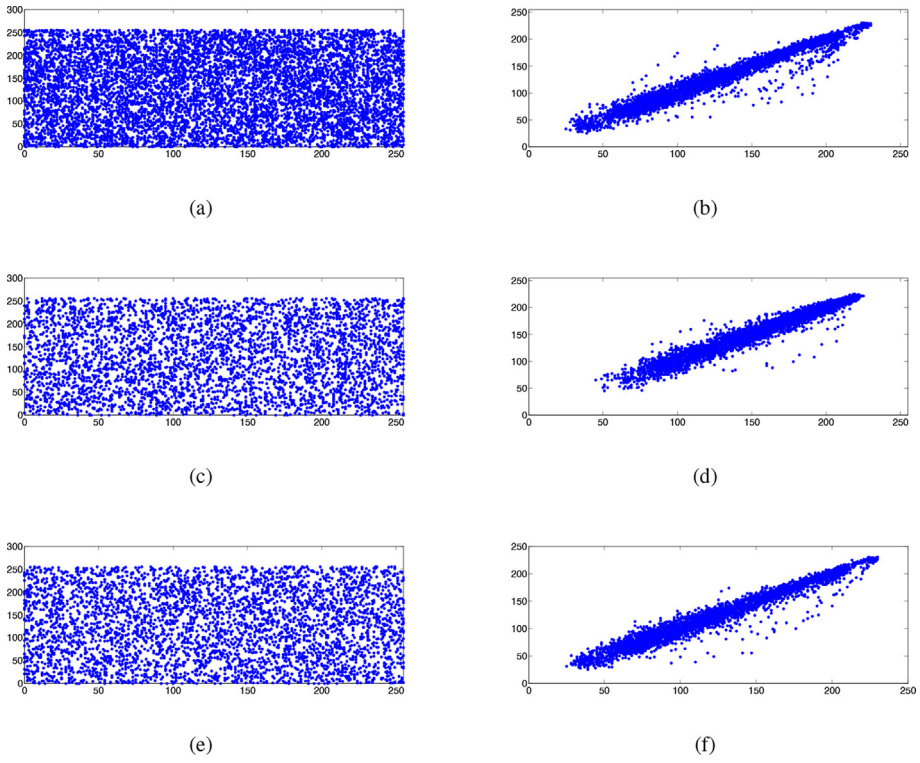



**Fig. 15.** Correlation coefficient of 5.1.10 image (a) Horizontal cipher image , (b) horizontal plain image, (c) vertical cipher image, (d) vertical plain image, (e) diagonal plain image, (f) and diagonal cipher image.

**Table 4**
Correlation coefficient of the encrypted plain image 5.1.9 with other algorithms.

| Image | Cipher image | | |
|---|---|---|---|
| | D | V | H |
| Proposed algorithm | −0.0003 | 0.0006 | −0.007 |
| Ref. (Hua and Zhou, 2016) | 0.0012 | 0.0011 | 0.007 |
| Ref. (Alawida et al., 2019a) | 0.0004 | 0.0021 | 0.005 |
| Ref. (Hua et al., 2018) | 0.0025 | 0.0078 | 0.022 |

The UACI test has two critical values, which are the strict range $U_{\alpha}^{-}$ and $U_{\alpha}^{+}$. To pass the UACI test, the image cipher's value must fall within these intervals. These critical values can be calculated for different images as follows:

$$\begin{cases} U_{\alpha}^{*-} = \mu_u - \Phi^{-1}(\alpha/2) \times \sigma_u; \\ U_{\alpha}^{*+} = \mu_u + \Phi^{-1}(\alpha/2) \times \sigma_u, \end{cases} \tag{23}$$

where

$$\mu_u = \frac{L+2}{3L+3} \tag{24}$$

and

$$\sigma_u^2 = \frac{(L+2)(L^2+2L+3)}{18(L+1)^2 \times L \times MN} \tag{25}$$

In the dataset, there are three different image sizes: 6 images with a size of $256 \times 256$, 18 images with a size of $512 \times 512$, and 4 images with a size of $1024 \times 1024$. The varying sizes of images result in different results because some proposed algorithms may perform well with larger sizes but poorly with smaller sizes. This highlights how the proposed algorithm performs with different sizes. The significance level is set to $\alpha = 0.05$ for all image sizes, to compare the critical values of NPCR and UACI. Using the same secret key, each plain image is encrypted twice, once before and once after flipping one bit. The results of the NPCR and UACI tests are calculated and compared for the proposed algorithm and three other algorithms in Table 5. The results of the proposed algorithm reinforce the previously mentioned analysis of the diffusion feature and plain text sensitivity. The proposed algorithm passes more than 25 images in both tests, demonstrating its resistance to the differential attack.

### 5.4. Local Shannon entropy

Local Shannon entropy (LSE) is a statistical measure that quantifies the amount of uncertainty or randomness present in a digital image. It is used to evaluate the randomness of the pixel values in an image and is calculated for each pixel in the image based on the probability distribution of the gray levels in its surrounding area. The entropy value is higher for pixels that have a high degree of randomness and lower for pixels with less randomness.

The pixel distribution in an encrypted image should be uniform not only across the entire image, but also in its non-overlapping sections. This is because any randomly selected part of the image should display the same uniform distribution as the whole image. To assess the randomness and distribution of a digital image, the LSE can be employed. The Shannon entropy for the image $I$ can be calculated as:

$$H(I) = -\sum_{j=0}^{L-1} p(I_j) log_2 p(I_j), \tag{26}$$

where $p(I_j)$ is the discrete probability density function (PDF), $I_j$ is the pixel value from 0 to 255, and $L$ is grey level ($L = 256$ for an 8-bit grey image).

The computation of LSE involves determining the randomness and uniform distribution of pixels in the image through the use

of the Shannon entropy measure on non-overlapping image blocks that are randomly selected and have a uniform number of pixels. The LSE is then calculated as the average of all Shannon entropy values of the randomly selected blocks as

$$H_{k,T_B}(I) = \sum_{i=1}^{k} \frac{H(I_{B_i})}{k}, \tag{27}$$

where $H(I_{B_i})$ denotes Shannon entropy of non-overlapping image blocks, $B_i$ is represented by $H(I_{B_i})$. The number of blocks and the number of pixels in each block are represented by $k$ and $T_B$ respectively. In order to pass the LSE test, the degree of LSE for the cipher image must be within the critical interval of (7.901515698, 7.903422936). This critical interval is determined using the significance value $\alpha = 0.001$ and the parameters $(k, TB) = (30, 1936)$. The theoretical value of LSE can be computed as 7.902469317.

The proposed algorithm was tested on 28 plain images along with three other algorithms for LSE. The results, listed in Table 6, showed that the encrypted images from other algorithms passed the LSE test at a maximum of around 85%, while the proposed algorithm had a passing rate of over 90%. This indicates that the proposed algorithm provides a high degree of randomness and uniform distribution in local sections of the encrypted image. This is due to the two-round encryption process using blocks of varying sizes, which contributes to the high randomness in the cipher image. The proposed algorithm faced challenges during the test with some images for two main reasons. Firstly, the size of the non-overlapping blocks was set at 1936 pixels, which made it difficult to achieve the ideal value for Shannon entropy. Secondly, the encrypted block required 7 occurrences for each gray scale to reach the ideal value, resulting in 1792 occurrences in total. This meant that 144 of the 256 gray scales occurred 8 times, while the remaining 112 occurred 7 times in the encrypted block. As a result, the final score for Shannon entropy did not reach the ideal value of 8 for the uniform block. Additionally, the algorithm struggled to achieve good randomness in the cipher image, which affected its ability to calculate the LSE based on the Shannon entropy of 30 blocks. The algorithm averaged the results to show scores within the critical interval, but unfortunately, three images failed to produce results within the interval. Specifically, between 17 and 20 blocks failed to achieve scores within the critical interval, impacting the final score of the entire image.

### 5.5. Contrast analysis

The discovery of patterns in the cipher image is another important metric that depends on the image's size and direction. To measure the local feature differences between pixels over the four directions in the cipher image, contrast analysis is employed. Mathematically, it can be expressed as:

$$C = \sum_{i,j} |i-j|^2 CO(i,j), \tag{28}$$

where $CO$ is co-occurrence matrix of the gray level index, $i$ and $j$ are 8-bit gray level images.

Statistically, $CO$ can be computed by tallying the fixed patterns in the image matrix, which depend on two factors: the distance between pixels within a pattern, and the four directions, namely horizontal (0), vertical (90), main diagonal (135), and secondary diagonal (45). In this experiment, both criteria were utilized to

**Table 5**
Comparison of NPCR and UACI scores for various images using different algorithms.

| Image | NPCR | | | | UACI | | | |
|---|---|---|---|---|---|---|---|---|
| | Proposed | Ref (Hua and Zhou, 2016) | Ref (Hua et al., 2018) | Ref (Alawida et al., 2019a) | Proposed | Ref (Hua and Zhou, 2016) | Ref (Hua et al., 2018) | Ref (Alawida et al., 2019a) |
| 256× 256 | $N^*_\alpha \geqslant 99.5693\%$ | | | | $U^{*-}_\alpha, U^{*+}_\alpha = (33.2824\%, 33.6447\%)$ | | | |
| 5.1.09 | 99.67 | 99.61 | 99.59 | 99.62 | 33.512 | 33.411 | 33.351 | 33.550 |
| 5.1.10 | 99.65 | 99.67 | 99.75 | 99.85 | 33.382 | 33.349 | 33.384 | 33.352 |
| 5.1.11 | 99.91 | 99.75 | 52.69 | 99.65 | 33.487 | 33.392 | 33.512 | 33.378 |
| 5.1.12 | 99.84 | 99.61 | 99.75 | 99.86 | 33.411 | 33.551 | 33.521 | 33.154 |
| 5.1.13 | 99.78 | 99.65 | 99.82 | 99.61 | 33.501 | 33.365 | 33.351 | 33.471 |
| 5.1.14 | 99.61 | 99.62 | 99.52 | 99.61 | 33.422 | 33.461 | 33.351 | 33.542 |
| 512×512 | $N^*_\alpha \geqslant 99.5893\%$ | | | | $U^{*-}_\alpha, U^{*+}_\alpha = (33.3730\%, 33.5541\%)$ | | | |
| 5.2.08 | 99.66 | 99.61 | 99.74 | 99.59 | 33.424 | 33.412 | 33.452 | 33.425 |
| 5.2.09 | 99.86 | 99.62 | 99.05 | 99.62 | 33.541 | 33.444 | 33.566 | 33.531 |
| 5.2.10 | 99.65 | 99.61 | 99.68 | 99.68 | 33.421 | 33.504 | 33.441 | 33.394 |
| 7.1.01 | 99.96 | 99.95 | 99.57 | 99.98 | 33.548 | 33.534 | 33.352 | 33.391 |
| 7.1.02 | 99.91 | 99.61 | 99.61 | 99.59 | 33.465 | 33.531 | 33.252 | 33.485 |
| 7.1.03 | 99.81 | 99.61 | 99.52 | 99.61 | 33.545 | 33.402 | 33.284 | 33.523 |
| 7.1.04 | 99.91 | 99.61 | 99.98 | 99.59 | 33.398 | 33.520 | 33.562 | 33.854 |
| 7.1.05 | 99.91 | 99.59 | 94.32 | 99.91 | 33.412 | 33.515 | 28.569 | 33.487 |
| 7.1.06 | 99.67 | 99.63 | 99.94 | 99.59 | 33.409 | 33.437 | 33.535 | 33.552 |
| 7.1.07 | 99.93 | 99.68 | 99.19 | 99.94 | 33.507 | 33.511 | 33.885 | 33.488 |
| 7.1.08 | 99.81 | 99.57 | 99.56 | 99.88 | 33.375 | 33.618 | 33.023 | 33.553 |
| 7.1.09 | 99.84 | 99.67 | 99.92 | 99.91 | 33.530 | 33.405 | 33.429 | 33.506 |
| 7.1.10 | 99.97 | 99.06 | 99.95 | 99.78 | 33.431 | 33.875 | 33.391 | 33.399 |
| boat.512 | 99.75 | 99.65 | 99.82 | 99.61 | 33.376 | 33.475 | 33.858 | 33.485 |
| elaine.512 | 99.74 | 99.62 | 60.88 | 99.94 | 33.378 | 33.440 | 32.589 | 33.457 |
| gray21.512 | 99.63 | 99.69 | 99.94 | 99.92 | 33.502 | 33.491 | 33.402 | 33.387 |
| numbers.512 | 99.61 | 99.61 | 99.63 | 99.93 | 33.389 | 33.589 | 33.851 | 33.158 |
| ruler.512 | 99.63 | 99.63 | 99.25 | 99.93 | 33.411 | 33.351 | 33.422 | 33.540 |
| 1024×1024 | $N^*_\alpha \geqslant 99.5994\%$ | | | | $U^{*-}_\alpha, U^{*+}_\alpha = (33.4183\%, 33.5088\%)$ | | | |
| 5.3.01 | 99.95 | 99.61 | 99.05 | 99.61 | 33.509 | 33.334 | 33.688 | 33.421 |
| 5.3.02 | 99.98 | 99.61 | 99.62 | 99.81 | 33.504 | 33.482 | 33.419 | 33.420 |
| 7.2.01 | 99.91 | 99.66 | 99.20 | 99.68 | 33.482 | 33.474 | 33.512 | 33.506 |
| testpat.1 k | 99.88 | 99.62 | 99.68 | 99.69 | 33.452 | 33.467 | 33.858 | 33.458 |
| Pass/All | 28/28 | 27/28 | 28/28 | 27/28 | 26/28 | 24/28 | 15/28 | 26/28 |

determine contrast values for various image sizes. For the purpose of comparison, the theoretical value of contrast features for a digital image was also calculated. The formula used to calculate the theoretical value is as follows: $C_{theoretical} = \sum_{i=1}^{256}\sum_{j=1}^{256} \frac{|i-j|^2}{(256)^2} = 10922.50$. This value assumes that all patterns have the same probability of occurring in an image.

The contrast analysis outcomes for the proposed algorithm and the three other algorithms are obtained and presented in Table 7. It can be observed that the proposed algorithm outperforms the other algorithms and achieves values closer to the ideal ones for most of the images in the dataset. The superiority of the proposed algorithm can be attributed to the novel approach utilized to provide the permutation operation. The employment of the improved logistic chaotic map with high randomness and ergodicity is reflected in the permutation operation in the first round of the proposed algorithm.

### 5.6. Key security

The secret key used in cryptography has two main components - keyspace and sensitivity. The keyspace is designed to protect against brute force attacks and should ideally be equal to or larger than $2^{128}$. The sensitivity of the key measures how effective it is in achieving the confusion feature. In this experiment, the key generation process is already discussed in Section 3.3.2, so it will not be including the keyspace component in this analysis.

To test the key sensitivity, a master secret key $K_1$ is created and another key $K_2$ is generated by flipping one bit. The same plain image (Lena) is encrypted using the two keys ($K_1 = FFFF\ FFFFFFFFFFFFFFFFFFFFFFFFFFFF$) and ($K_2 = FFFFFFFFFFFFFFFFFFFFFFFF\ FFFFFFFE$) to generate two encrypted images ($C_1$ and $C_2$), respectively. The difference between $C_1$ and $C_2$ is then calculated. Additionally, the second cipher image $C_2$ is decrypted by the first key $K_1$, and the first cipher image $C_1$ is decrypted by the second key $K_2$. All these results are shown in Fig. 16.

The results in Fig. 16 demonstrate the proposed algorithm's high sensitivity to one-bit changes in the key, resulting in cipher images that cannot be decrypted back to the original plain image. Several reasons contribute to this sensitivity, including the perturbed logistic map's high sensitivity to small changes in chaotic variables, which is the first reason. The second reason is that a one-bit change in the secret key will cause changes to the chaotic variables, leading to a completely different chaotic data sequence. Finally, the chaotic sequence has two significant impacts on image pixels: permutation and substitution, both of which depend on the chaotic sequence.

### 5.7. Speed analysis

Another important measure in cipher design is speed analysis, which can be a crucial factor. A faster algorithm is necessary for handling large amounts of data, such as images, in a short amount of time and with low resources. Increasing the number of rounds in a cipher can lead to higher security, but slower performance. Conversely, decreasing the number of rounds can improve performance but lower security. To develop a high-performance algorithm with high-quality security, each step in the algorithm design should be thoroughly studied and selected from two perspectives; security and speed.

This algorithm takes care of the performance during its generation. A 1D perturbed chaotic logistic map is used to generate a data sequence as a pre-processing step, which can be used for multiple images. The substitution operation is only performed in two rounds through the simple logical operation of Xoring. A lightweight hash function is used to generate 64 bits very quickly, and it is used only 512 times for small data over different image sizes. Therefore, the proposed algorithm is capable of encrypting images quickly. The encryption speed of the proposed algorithm is compared to that of other algorithms using various sizes of plain images in Table 8. All the algorithms included in this test were reimplemented on the same machine multiple times, and the aver-

**Table 6**
Comparison of LSE scores across different algorithms.

| Image | Proposed Algorithm | Ref (Hua and Zhou, 2016) | Ref (Hua et al., 2018) | Ref (Alawida et al., 2019a) |
|---|---|---|---|---|
| 5.1.09 | **7.902822** | **7.903395** | **7.902831** | **7.902758** |
| 5.1.10 | **7.903039** | **7.903271** | **7.903056** | **7.902215** |
| 5.1.11 | 7.903426 | **7.903043** | 7.904225 | **7.901642** |
| 5.1.12 | **7.903298** | 7.903468 | **7.903068** | **7.902335** |
| 5.1.13 | **7.902854** | **7.902728** | **7.903015** | 7.900207 |
| 5.1.14 | **7.903213** | **7.902795** | **7.903077** | **7.902531** |
| 5.2.08 | **7.903246** | **7.902831** | 7.903439 | **7.902487** |
| 5.2.09 | **7.902445** | **7.903028** | 7.904521 | **7.902681** |
| 5.2.10 | **7.902207** | 7.903511 | **7.902562** | **7.902152** |
| 7.1.01 | **7.902641** | **7.903362** | 7.903452 | **7.901772** |
| 7.1.02 | **7.903242** | **7.902706** | 7.903521 | **7.902483** |
| 7.1.03 | **7.902521** | **7.903252** | **7.903262** | **7.902725** |
| 7.1.04 | 7.901223 | **7.903313** | 7.901251 | 7.893536 |
| 7.1.05 | **7.903356** | **7.903103** | **7.902256** | 7.900743 |
| 7.1.06 | **7.903315** | **7.902625** | **7.902154** | **7.902163** |
| 7.1.07 | 7.903451 | **7.902435** | **7.903132** | **7.902208** |
| 7.1.08 | **7.901895** | **7.903675** | **7.902876** | **7.903055** |
| 7.1.09 | **7.903123** | **7.902813** | 7.901251 | **7.902832** |
| 7.1.10 | **7.902156** | **7.904668** | **7.903298** | **7.902407** |
| boat.512 | **7.902589** | **7.902632** | 7.903556 | **7.902674** |
| elaine.512 | **7.902556** | **7.902486** | **7.901542** | **7.902699** |
| gray21.512 | **7.903056** | **7.902543** | **7.902452** | **7.901923** |
| numbers.512 | **7.903021** | **7.902885** | **7.903214** | **7.902488** |
| ruler.512 | **7.902450** | 7.903805 | **7.903238** | **7.903052** |
| 5.3.01 | **7.902892** | **7.903106** | **7.903486** | 7.887108 |
| 5.3.02 | **7.901452** | **7.903263** | **7.901875** | **7.902447** |
| 7.2.01 | **7.903226** | **7.902848** | **7.902025** | 7.898703 |
| testpat.1 k | **7.900125** | 7.901336 | **7.903195** | **7.902771** |
| {Pass/All} | 25/28 | 24/28 | 20/28 | 22/28 |

**Table 7**
Contrast values for different image encryption algrithms.

| Image | Proposed Algorithm | Ref (Hua and Zhou, 2016) | Ref (Hua et al., 2018) | Ref (Alawida et al., 2019a) |
|---|---|---|---|---|
| 5.1.09 | 10923.5 | 10883.87 | 10918.65 | 10922.25 |
| 5.1.10 | 10928.36 | 11012.40 | 10971.56 | 10922.32 |
| 5.1.11 | 10929.12 | 10992.52 | 10892.87 | 10891.5 |
| 5.1.12 | 10918.02 | 10920.51 | 10906.51 | 10905.66 |
| 5.1.13 | 10920.45 | 10881.36 | 10918.48 | 10918.01 |
| 5.1.14 | 10908.25 | 10902.43 | 10935.12 | 10925.21 |
| 5.2.08 | 10929.25 | 10919.69 | 10930.46 | 10922.67 |
| 5.2.09 | 10902.25 | 10943.73 | 10912.54 | 10945.07 |
| 5.2.10 | 10931.25 | 10925.33 | 10894.26 | 10899.12 |
| 7.1.01 | 10922.61 | 10929.88 | 10869.71 | 10885.62 |
| 7.1.02 | 10932.26 | 10938.70 | 10925.80 | 10923.56 |
| 7.1.03 | 10916.23 | 10938.56 | 10934.25 | 10917.18 |
| 7.1.04 | 10930.21 | 10937.08 | 10927.72 | 10935.25 |
| 7.1.05 | 10925.25 | 10929.65 | 10944.33 | 10924.52 |
| 7.1.06 | 10924.12 | 10928.22 | 10908.84 | 10909.12 |
| 7.1.07 | 10919.12 | 10923.96 | 10939.20 | 10919.25 |
| 7.1.08 | 10918.22 | 10895.11 | 10879.07 | 10899.94 |
| 7.1.09 | 10909.12 | 10902.03 | 10901.27 | 10905.26 |
| 7.1.10 | 10908.22 | 10902.46 | 10959.93 | 10968.25 |
| boat.512 | 10924.25 | 10920.95 | 10918.76 | 10928.5 |
| elaine.512 | 10921.92 | 10879.22 | 10913.04 | 10928.25 |
| gray21.512 | 10924.25 | 10918.86 | 10932.86 | 10922.5 |
| numbers.512 | 10919.35 | 10926.62 | 10922.22 | 10925.21 |
| ruler.512 | 10911.23 | 10904.14 | 10915.25 | 10975.22 |
| 5.3.01 | 10908.25 | 10928.71 | 10931.88 | 10922.48 |
| 5.3.02 | 10924.25 | 10918.22 | 10919.96 | 10918.02 |
| 7.2.01 | 10924.25 | 10929.86 | 10911.25 | 10914.25 |
| testpat.1 k | 10923.98 | 10912.45 | 10908.88 | 10919.22 |
| {Mean} | 10920.62 | 10923.09 | 10919.45 | 10921.02 |
| {Std} | 7.963 | 28.50 | 21.72 | 19.1293 |

age values were calculated. It is observed that the proposed scheme has a higher encryption speed compared to the other algorithms.

### 5.8. Performance comparison

The presented table, Table 9, compares the performance of various image encryption algorithms based on several measurements including correlation coefficient, Shannon entropy, NPCR, UACI, keyspace, and speed. The comparison results were obtained using the Lena image for all algorithms. The proposed algorithm has a correlation coefficient of 0.0004, Shannon entropy of 7.9994, NPCR of 99.6125, UACI of 33.4525, keyspace of $2^{128}$, and speed of 0.25. The NPCR and UACI measurements for the proposed algorithm are better than most other algorithms. However, the speed of the proposed algorithm is not the fastest due to device specifications and the number of algorithms running. Nevertheless, the proposed algorithm shows strong security properties and comparability with

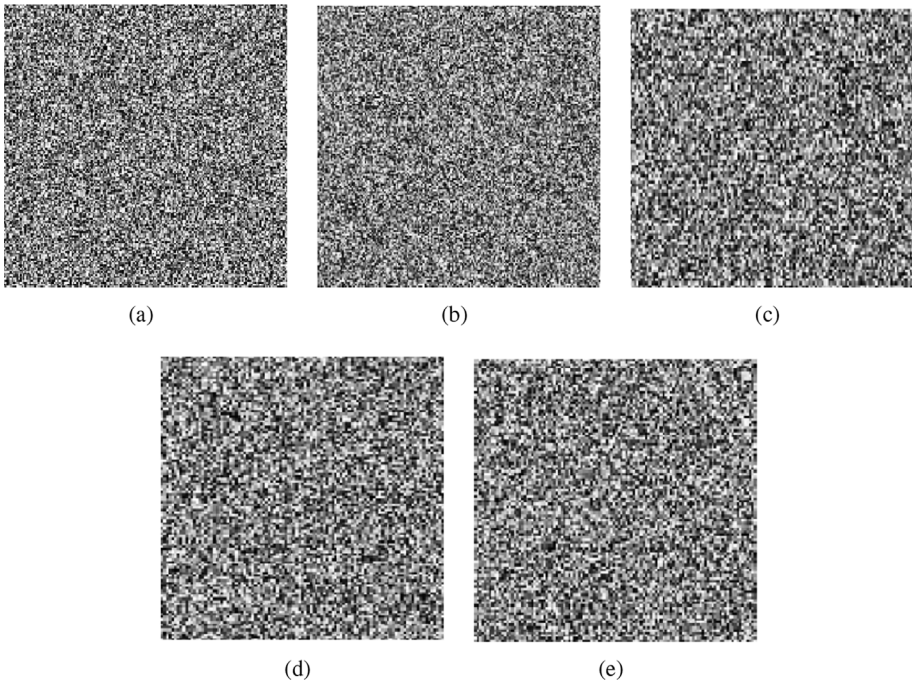



**Fig. 16.** Key sensitivity analysis (a) Cipherimage C1 by K1 , (b) Cipher image C2 by K2, (c) Difference between C1 and C2, (d) Decrypt C1 by K2, (e) decrypt C2 by K1.

other algorithms. Its fast permutation can also be used as a pre-process on many digital images.

### 5.9. Discussion

The proposed algorithm aims to efficiently encrypt images of various sizes while ensuring high sensitivity to secret keys. To achieve this, the logistic chaotic map is enhanced using the perturbation chaotification method in the first stage. A hybrid of the forward and backward perturbation methods is applied to generate a high-quality chaotic map capable of generating data sequences with superior security. The improved chaotic map is utilized to achieve diffusion and confusion in the image cipher through a novel approach for permutation and a straightforward substitution method.

The proposed image encryption algorithm boasts significant features, which are enumerated below:

- A secret key, which is not related to the plain image, is employed to generate chaotic variables. A single secret key can be used to encrypt multiple images.
- The key generation, chaotic data sequence, and block generation can be pre-processed for encryption, thereby increasing the algorithm's performance and enabling it to be applied to many images.
- The proposed image encryption technique is capable of processing various image sizes and types.
- The perturbed logistic map is employed to generate high-complexity chaotic sequences, which are more sophisticated than those produced by a classical logistic map.
- The chaotic data sequence is responsible for the permutation and substitution operations, thereby empowering the encryption algorithm with robust security capabilities.
- The non-overlapping blocks are dynamic in size and are based on the chaotic data sequence.
- The novel permutation operation is contingent upon the random, ergodic, and uniform distribution of the improved logistic map.
- The lightweight hash function is incredibly fast and produces 64-bit values for each block's average. It is utilized to achieve diffusion with highly efficient characteristics.
- The proposed algorithm exhibits high sensitivity to a one-bit alteration in the secret key and plain image.
- The cipher is resilient against well-known security attacks, such as statistical and differential attacks, as elaborated in the following section.

A chosen plaintext attack (CPA) or known-plaintext attack (KPA) is a cryptanalytic technique used to break encryption. In a CPA, the attacker chooses the plaintext messages and observes their corresponding ciphertexts. In a KPA, the attacker knows some of the plaintext-ciphertext pairs. These attacks are particularly powerful because they allow the attacker to gain insights into the encryption process, such as the secret key or the encryption algorithm itself. If the encryption algorithm is weak or the key is poorly chosen, the attacker may be able to derive the key or the original plaintext message from the known or chosen ciphertexts. To prevent these attacks, there is a set variety of techniques, such as ensuring the encryption algorithm is strong and ensuring that the key is kept secret and frequently rotated.

Consider three types of plain images: full black, full white, and half black and white. The black image consists of pixels with zero values, while the white image has 255 values of gray level. During the first step of encryption's permutation operation, the plain image remains unchanged. In the second step, substitution, the 8-bit chaotic number plays a crucial role in updating the pixel values. All results are updated based on the 8-bit chaotic number. After two rounds, the cipher image has completely new pixel values with high randomness. The same process is repeated for white or half images, and the 8-bit chaotic points are Xored with the previously encrypted pixels and previous hashed block to generate a new, different image. This process helps to ensure that the encrypted images are highly randomized and resistant to decryption by unauthorized parties.

It is crucial to acknowledge the limitations of the proposed encryption algorithm in order to understand its weaknesses. Firstly, the hashed blocks used in the encryption process are highly susceptible to minor alterations, which may introduce unwanted noise during decryption. To address this, the algorithm incorporates XOR operations between the hash value of the encrypted block, the first 8 pixels of the following blocks, and the 8 numbers of the chaotic data sequence. Although the impact of sensitive hashing is limited to one pixel at a time, it still persists and grows as the decryption process continues. Secondly, the number of available secret keys is constrained by the number of chaotic variables. Finally, the conversion process from floating point numbers to 8-bit numbers reduces the encryption speed.

**Table 8**
Encryption time (second) with different image sizes and encryption algorithms.

| Image Size | Proposed Algorithm | Ref (Hua and Zhou, 2016) | Ref (Hua et al., 2018) | Ref (Alawida et al., 2019a) |
|---|---|---|---|---|
| $128 \times 128$ | **0.0156** | 0.1582 | 0.0221 | 0.1131 |
| $256 \times 256$ | **0.0625** | 0.2594 | 0.0894 | 0.4525 |
| $512 \times 512$ | **0.2500** | 18.152 | 0.3576 | 1.81 |
| $1024 \times 1024$ | **0.9845** | 79.25 | 1.5124 | 7.24 |
| Average | **0.3281** | 24.4549 | 0.4954 | 2.4039 |

**Table 9**
Performance comparison of various image encryption algorithms.

| Algorithms | Comparison measurements | | | | | |
|---|---|---|---|---|---|---|
| | Correlation coefficient | Shannon Entropy | NPCR | UACI | Keyspace | Speed (S) |
| Ref. (Azam et al., 2023) | −0.0016 | 7.9993 | 99.6075 | 33.4687 | $2^{100}$ | 1.0128 |
| Ref. (Erkan et al., 2022) | −0.000021 | 7:9993 | 99.6084 | 33.4714 | $2^{298}$ | 0.2984 |
| Ref. (Teng et al., 2022) | −0.001627 | 7.9914 | 99.5937 | 33.4086 | $2^{512}$ | 1.769703 |
| Ref. (Zhou et al., 2023) | −0.0001 | 7.9994 | 99.60 | 33.42 | $2^{448}$ | - |
| Ref. (Gao et al., 2023c) | −0.0001 | 7.9992 | 99.6102 | 33.4465 | $2^{256}$ | 0.2205 |
| Proposed Algorithm | 0.0004 | 7.9994 | 99.6125 | 33.4525 | $2^{128}$ | 0.25 |

The proposed algorithm has some areas of improvement that warrant further research. Firstly, the secret key is currently limited to 128 bits. To increase the strength of the encryption, additional chaotic variables can be added by incorporating more initial conditions. Secondly, the proposed algorithm exhibits high sensitivity to minor changes in the cipher image, owing to the use of the hash function and non-overlapping block method. Even the slightest alteration in the encrypted pixel can result in exponential growth in the plaintext, which is a limitation that requires attention.

## 6. Conclusion

In conclusion, this paper has proposed a new image encryption method based on digital chaos, which provides shared properties of randomness and sensitivity to ensure secure encryption. The proposed method enhances the logistic chaotic map by combining backward and forward methods, resulting in a new perturbed chaotic map with better security than existing chaotic systems. The chaotic performance of the new method, including chaotic range, complex behavior, ergodicity, and sensitivity, was improved compared to other methods. The proposed algorithm uses a novel permutation operation to achieve diffusion and confusion features, which are pre-processed. In the encryption process, two substitutions were done during two rounds to generate a cipher image. The resulting algorithm provides high security and low time consumption, withstanding various statistical and cyber attacks. Overall, this paper's contributions provide a significant advancement in image encryption algorithms, offering a secure and efficient method for protecting digital images during transmission. Future work could involve investigating the application of the proposed perturbed logistic map in other multimedia data types, including text, drone images, remote sensing images, video streaming, and video conferencing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work was fully supported by Abu Dhabi University under Grant No. 19300786.
The author expresses gratitude to the Office of Research & Sponsored Programs at Abu Dhabi University in the United Arab Emirates for their support of this research article.

## References

Alawida, M., Samsudin, A., Teh, J.S., Alkhawaldeh, R.S., 2019a. A new hybrid digital chaotic system with applications in image encryption. Signal Process. 160, 45–58.
Alawida, M., Samsudin, A., Teh, J.S., Alshoura, W., 2019b. Deterministic chaotic finite-state automata. Nonlinear Dyn. 98 (3), 2403–2421.
Alawida, M., Teh, J.S., Samsudin, A., Alshoura, W.H., 2019c. An image encryption scheme based on hybridizing digital chaos and finite state machine. Signal Process. 164, 249–266. https://doi.org/10.1016/j.sigpro.2019.06.013. URL https://www.sciencedirect.com/science/article/pii/S0165168419302221.
Alawida, M., Teh, J.S., Oyinloye, D.P., Alshoura, W.H., Ahmad, M., Alkhawaldeh, R.S., 2020. A new hash function based on chaotic maps and deterministic finite state automata. IEEE Access 8, 113163–113174. https://doi.org/10.1109/ACCESS.2020.3002763.
Alawida, M., Samsudin, A., Alajarmeh, N., Teh, J.S., Ahmad, M., Alshoura, W.H., 2021. A novel hash function based on a chaotic sponge and dna sequence. IEEE Access 9, 17882–17897. https://doi.org/10.1109/ACCESS.2021.3049881.
Alawida, M., Teh, J.S., Mehmood, A., Shoufan, A., et al., 2022. A chaos-based block cipher based on an enhanced logistic map and simultaneous confusion-diffusion operations. J. King Saud Univ.-Comput. Informat. Sci. 34 (10), 8136–8151.
Alawida, M., Teh, J.S., Alshoura, W., et al., 2023. A new image encryption algorithm based on dna state machine for uav data encryption. Drones 7 (1), 38.
Aumasson, J.-P., Henzen, L., Meier, W., Naya-Plasencia, M., 2013. Quark: A lightweight hash. J. Cryptol. 26, 313–339.
Azam, N.A., Zhu, J., Hayat, U., Shurbevski, A., 2023. Towards provably secure asymmetric image encryption schemes. Inf. Sci. 631, 164–184.
Aziz, H., Gilani, S.M.M., Hussain, I., Abbas, M.A., 2020. A novel symmetric image cryptosystem resistant to noise perturbation based on s8 elliptic curve s-boxes and chaotic maps. Eur. Phys. J. Plus 135 (11), 907.
Bezerra, J.I.M., de Almeida Camargo, V.V., Molter, A., 2021. A new efficient permutation-diffusion encryption algorithm based on a chaotic map. Chaos, Solitons Fractals 151, 111235.
Cao, L.-C., Luo, Y.-L., Qiu, S.-H., Liu, J.-X., 2015. A perturbation method to the tent map based on lyapunov exponent and its application. Chin. Phys. B 24 (10), 100501. https://doi.org/10.1088/1674-1056/24/10/100501.
Chai, X., Chen, Y., Broyde, L., 2017. A novel chaos-based image encryption algorithm using dna sequence operations. Opt. Lasers Eng. 88, 197–213.
Enayatifar, R., Guimarães, F.G., Siarry, P., 2019. Index-based permutation-diffusion in multiple-image encryption using dna sequence. Opt. Lasers Eng. 115, 131–140.
Erkan, U., Toktas, A., Toktas, F., Alenezi, F., 2022. 2d ep-map for image encryption. Inf. Sci. 589, 770–789. https://doi.org/10.1016/j.ins.2021.12.126. URL https://www.sciencedirect.com/science/article/pii/S0020025521013475.
Fan, C., Ding, Q., 2019. Effects of limited computational precision on the discrete chaotic sequences and the design of related solutions. Complexity 2019, 1–10. https://doi.org/10.1155/2019/3510985.
Fang, P., Liu, H., Wu, C., Liu, M., 2022. A survey of image encryption algorithms based on chaotic system. Visual Comput., 1–29
Gao, S., Wu, R., Wang, X., Liu, J., Li, Q., Tang, X., 2023a. Efr-cstp: Encryption for face recognition based on the chaos and semi-tensor product theory. Inf. Sci. 621, 766–781.
Gao, S., Wu, R., Wang, X., Wang, J., Li, Q., Wang, C., Tang, X., 2023b. A 3d model encryption scheme based on a cascaded chaotic system. Signal Process. 202, 108745. https://doi.org/10.1016/j.sigpro.2022.108745. URL https://www.sciencedirect.com/science/article/pii/S0165168422002845.
Gao, S., Wu, R., Wang, X., Liu, J., Li, Q., Wang, C., Tang, X., 2023c. Asynchronous updating boolean network encryption algorithm. IEEE Trans. Circ. Syst. Video Technol.
Grassbergert, P., Procaccia, T., 1983. Characterization of strange attractors. Phys. Rev. Lett 50 (5), 346–349.
Han, C., 2019. An image encryption algorithm based on modified logistic chaotic map. Optik 181, 779–785.
Hoang, T.M., 2022. A novel design of multiple image encryption using perturbed chaotic map. Multimedia Tools Appl. 81 (18), 26535–26589.
Hua, Z., Zhou, Y., 2016. Image encryption using 2d logistic-adjusted-sine map. Inf. Sci. 339, 237–253. https://doi.org/10.1016/j.ins.2016.01.017.
Hua, Z., Zhou, Y., 2018. One-dimensional nonlinear model for producing chaos. IEEE Trans. Circuits Syst. I Regul. Pap. 65 (1), 235–246. https://doi.org/10.1109/tcsi.2017.2717943.
Hua, Z., Jin, F., Xu, B., Huang, H., 2018. 2d logistic-sine-coupling map for image encryption. Signal Process. 149, 148–161.
Hua, Z., Zhou, Y., Huang, H., 2019. Cosine-transform-based chaotic system for image encryption. Inf. Sci. 480, 403–419. https://doi.org/10.1016/j.ins.2018.12.048.
Hu, H., Deng, Y., Liu, L., 2014. Counteracting the dynamical degradation of digital chaos via hybrid control. Commun. Nonlinear Sci. Numer. Simul. 19 (6), 1970–1984. https://doi.org/10.1016/j.cnsns.2013.10.031.
Jun, Z., Hanping, H., Xiang, X., 2018. Applications of symbolic dynamics in counteracting the dynamical degradation of digital chaos. Nonlinear Dyn. 94 (2), 1535–1546.
Li, C., Luo, G., Li, K.Q.C., 2017. An image encryption scheme based on chaotic tent map. Nonlinear Dyn. 84 (1), 127–133.
Liu, L., Lin, J., Miao, S., Liu, B., 2017. A double perturbation method for reducing dynamical degradation of the digital baker map. Int. J. Bifurcat. Chaos 27 (07), 1750103.
Liu, Y., Luo, Y., Song, S., Cao, L., Liu, J., Harkin, J., 2017. Counteracting dynamical degradation of digital chaotic chebyshev map via perturbation. Int. J. Bifurcat. Chaos 27 (03).
Liu, L., Xiang, H., Li, X., 2021. A novel perturbation method to reduce the dynamical degradation of digital chaotic maps. Nonlinear Dyn. 103, 1099–1115.
Luo, Y., Liu, Y., Liu, J., Tang, S., Harkin, J., Cao, Y., 2021. Counteracting dynamical degradation of a class of digital chaotic systems via unscented kalman filter and perturbation. Inf. Sci. 556, 49–66.
Mansouri, A., Wang, X., 2020. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. Inf. Sci. 520, 46–62.
Mansouri, A., Wang, X., 2021. Image encryption using shuffled arnold map and multiple values manipulations. Visual Comput. 37 (1), 189–200.
Muthu, J.S., Murali, P., 2022. A novel dicom image encryption with jsmp map. Optik 251, 168416.
Noshadian, S., Ebrahimzade, A., Kazemitabar, S.J., 2020. Breaking a chaotic image encryption algorithm. Multimedia Tools Appl. 79, 25635–25655.
Shahna, K., Mohamed, A., 2021. Novel hyper chaotic color image encryption based on pixel and bit level scrambling with diffusion. Signal Process.: Image Commun. 99, 116495.
Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Technical J. 27 (3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
Talhaoui, M.Z., Wang, X., 2021. A new fractional one dimensional chaotic map and its application in high-speed image encryption. Inf. Sci. 550, 13–26. https://doi.

org/10.1016/j.ins.2020.10.048. URL https://www.sciencedirect.com/science/article/pii/S0020025520310380.
Teng, L., Wang, X., Xian, Y., 2022. Image encryption algorithm based on a 2d-clss hyperchaotic map using simultaneous permutation and diffusion. Inf. Sci. 605, 71–85.
The usc-sipi image database, http://sipi.usc.edu/database/database.php.
Wafa' Hamdan, Z.Z., Alshoura, Teh, J.S., Alawida, M., 2022. An fpp-resistant svd-based image watermarking scheme based on chaotic control. Alexandria Eng. J. 61 (7), 5713–5734.
Wang, X., Gao, S., 2020. Image encryption algorithm for synchronously updating boolean networks based on matrix semi-tensor product theory. Inf. Sci. 507, 16–36.
Wang, X., Zhang, M., 2021. An image encryption algorithm based on new chaos and diffusion values of a truth table. Inf. Sci. 579, 128–149. https://doi.org/10.1016/j.ins.2021.07.096. URL https://www.sciencedirect.com/science/article/pii/S0020025521007921.
Wang, X., Su, Y., Liu, L., Zhang, H., Di, S., 2021. Color image encryption algorithm based on fisher-yates scrambling and dna subsequence operation. Visual Comput., 1–16
Wang, W., Han, Z., Alazab, M., Gadekallu, T.R., Zhou, X., Su, C., 2022. Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps. IEEE Trans. Ind. Appl. 58 (5), 5616–5623.
Wu, Y., Noonan, J.P., Agaian, S., 2011. NPCR and UACI randomness tests for image encryption. J. Sel. Areas Telecommun. 4 (1).
Wu, X., Zhu, B., Hu, Y., Ran, Y., 2017. A novel colour image encryption scheme using rectangular transform-enhanced chaotic tent maps. IEEE Access. https://doi.org/10.1109/access.2017.2692043. 1-1.
Yu, F., Li, L., Tang, Q., Cai, S., Song, Y., Xu, Q., 2019. A survey on true random number generators based on chaos. Discr. Dyn. Nat. Soc. 2019, 1–10.
Zhang, Y., Chen, A., Tang, Y., Dang, J., Wang, G., 2020. Plaintext-related image encryption algorithm based on perceptron-like network. Inf. Sci. 526, 180–202.
Zheng, J., Hu, H., Xia, X., 2018. Applications of symbolic dynamics in counteracting the dynamical degradation of digital chaos. Nonlinear Dyn. 94 (2), 1535–1546. https://doi.org/10.1007/s11071-018-4440-6.
Zhongyun Hua, Y.Z., 2018. Shuang Yi, Medical image encryption using high-speed scrambling and pixel adaptive diffusion. Signal Process. 149, 148–161.
Zhou, S., Wang, X., Zhang, Y., 2023. Novel image encryption scheme based on chaotic signals with finite-precision error. Inf. Sci. 621, 782–798.
Zhu, C., Sun, K., 2018. Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps. IEEE Access 6, 18759–18770. https://doi.org/10.1109/access.2018.2817600.