

A Lightweight Chaos-Based RGB Image Encryption Scheme for IoT Using Dynamic Diagonal Strip Permutation and XOR Diffusion

Abdul Rehman Altaf^{1,2}, Hairulnizam Mahdin², Awais Mahmood³, Heru Nurwaristo⁴, Muhammad Hassan⁵, Abdullah Altaf²

¹Johns Hopkins Aramco Healthcare (JHAH), Dhahran 34465, Saudi Arabia

²Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia

³Faculty of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia

⁴Faculty of Computer Science, University of Brawijaya, Malang, Indonesia

⁵Arabian Drilling Company, Al-Khobar 34434, Saudi Arabia

E-mail: hi210007@student.uthm.edu.my, hairuln@uthm.edu.my, mawais@ksu.edu.sa, heru@ub.ac.id, hassany5415@gmail.com

Keywords: Image encryption, internet of things (IoT), chaotic maps, RGB images, lightweight encryption

Received: August 16, 2026

Internet of Things (IoT) devices have found applications in numerous areas, including smart homes, health-care monitoring, smart metering, waste management, and industrial systems. These devices frequently collect and transmit data to fog nodes or cloud platforms for further processing. Ensuring the security of such data remains challenging, particularly when resource-constrained devices are involved. This paper presents a lightweight chaos-based image encryption scheme based on a two-stage confusion–diffusion structure. The proposed method employs the three-dimensional Intertwining Logistic Map (ILM) to generate chaotic sequences that control the encryption operations. In the confusion stage, the input RGB image is decomposed into its red, green, and blue components, from which two color components are dynamically selected according to the generated chaotic sequences. Diagonal strips of varying positions and directions are then repeatedly exchanged between the selected components to disrupt both the spatial arrangement and cross-channel distribution of the image pixels. In the diffusion stage, a chaos-generated mask is combined with the scrambled image through bitwise XOR operations to modify the pixel intensities while maintaining computational simplicity. The proposed scheme is evaluated through a range of statistical and security analyses. Experimental results yield an average information entropy of 7.9975, close to the ideal value of 8 for an 8-bit image, while an average encryption time of 0.4269 seconds is obtained under the specified conventional computing environment. These results demonstrate the effectiveness and computational feasibility of the proposed approach and indicate its potential for resource-conscious image security applications. Validation on resource-constrained IoT and embedded platforms remains an important direction for future investigation.

Povzetek: Članek predstavlja računsko učinkovito metodo šifriranja slik za potencialno uporabo v napravah interneta stvari (IoT), ki temelji na kaotičnem preslikavanju in dvostopenjskem postopku zmede in difuzije ter z visoko informacijsko entropijo in kratkim časom šifriranja izkazuje obetavne rezultate pri zagotavljanju varnosti slikovnih podatkov.

1 Introduction

As the Internet of Things (IoT) continues to gain widespread recognition, a secured transmission of information within the IoT ecosystem has attained an increasingly vital importance. IoT technology has seamlessly integrated into our everyday lives, with applications ranging from health-monitoring smart bands to smart home systems. Government organizations are investing substantial resources in endeavors like smart grid infrastructure, smart traffic management, and smart cities, while industries are already implementing smart factories and smart grid sys-

tems. These IoT devices play a pivotal role in sensing, storing, and transmitting data, much of which can be highly sensitive and, therefore, requires utmost protection. Securing data on IoT nodes and safeguarding its transmission from source to destination against potential intruders is of paramount importance. One straightforward approach to achieve this security is by encrypting data immediately after its sensing or collection within the IoT device itself. Several encryption algorithms have been developed for this purpose, including the International Data Encryption Algorithm (IDEA) [12], the Data Encryption Standard (DES) [4], and the Advanced Encryption Standard (AES)

[1]. However, their applicability in IoT devices is limited due to the significant computational overhead and extensive time requirements associated with decryption and encryption processes. Moreover, these algorithmic techniques have been found to be less suitable to encrypt large volumes of data, like images.

Encrypting data for IoT devices necessitates an encryption algorithm with minimal computational time. Chaotic encryption techniques have emerged as promising candidates for achieving security with a reduced computational overhead. These techniques/algorithms typically comprise two key phases: confusion (scrambling) and diffusion (changing intensity values) [38]. During the phase of scrambling, pixels' positions in data or image are shuffled to render secret information appear meaningless. Conversely, diffusion phase involves replacing pixel intensity values in data or image with alternative values. Encryption schemes that rely solely on permutation [37] are not sufficiently robust because they only alter pixel positions while leaving the histogram unchanged, making them susceptible to compromise. To enhance encryption algorithms' security, a diffusion phase is combined with the confusion phase [10].

In an era dominated by digital communication and multimedia applications, the security of visual data, particularly images, has emerged as a paramount concern. The ubiquitous use of images in fields ranging from healthcare to social media underscores the need for robust cryptographic solutions to safeguard these visual assets. Traditional cryptographic techniques have proven effective for securing textual data; however, they face significant challenges when applied to images [20]. Cryptography, the science of secure communication, has long been the cornerstone of data protection [2]. Historically, cryptographic methods have been tailored to secure textual data, and these techniques have thrived in conventional computing environments. Nevertheless, the landscape of information exchange has evolved dramatically, with digital imagery becoming a dominant mode of communication. It is becoming more and more clear in this context that specialized cryptographic solutions are required for images.

By solving the drawbacks of conventional cryptographic techniques when used with images, the rapidly expanding discipline of lightweight cryptography has gained popularity [39]. A paradigm change in the cryptographic field, lightweight cryptography focuses on creating effective and resource-efficient algorithms that are appropriate for the limitations of contemporary devices and applications. Specifically, lightweight image cryptography aims to close the gap between secure data processing and effective handling of visual input [13].

Images pose unique challenges for cryptographic protection. Their voluminous data size, real-time processing requirements, and diverse applications make them a formidable domain for cryptographic innovation [11]. Conventional cryptographic methods, which often demand extensive computational resources, are ill-suited for these

characteristics. Consequently, there is a pressing need for cryptographic techniques that can secure images without compromising their utility. In light of these challenges, this research endeavors to advance the field of lightweight image cryptography. Our work aims to develop novel cryptographic algorithms tailored specifically for images, prioritizing both security and efficiency. By addressing the limitations of existing cryptographic techniques, our research seeks to empower applications that rely on secure image transmission, storage, and authentication.

The domain of image cryptography stands at the crossroads of security, mathematics, and digital imaging. One captivating avenue within this realm is the integration of chaotic maps, a branch of mathematics that explores the deterministic yet unpredictable behavior of nonlinear dynamical systems [15]. Chaotic maps have emerged as a potent tool for lifting security of image encryption algorithms, owing to their intrinsic complexity and sensitivity to initial conditions [33].

Chaotic systems/maps are characterized by their sensitive dependence on initial conditions, often referred to as the "butterfly effect." A minuscule alteration in the starting point of a chaotic system can lead to vastly different trajectories, rendering prediction nearly impossible. This inherent unpredictability aligns seamlessly with the core tenets of cryptography, where security hinges on the inability to deduce meaningful information from encrypted data without the proper decryption key. In the context of image cryptography, chaotic maps serve as the driving force behind key generation and substitution-permutation operations. By employing chaotic dynamics, cryptographic algorithms can generate highly random and unpredictable sequences of numbers. These sequences, often referred to as keystreams, are then used to encrypt and decrypt the image data.

A lot of lightweight image ciphers have already been developed by the cryptographers to provide the security to the digital images. For instance, the work [16] wrote a novel algorithm in edge computing to provide the security to the images residing in the IoT devices in the cloud. This work fused the short comparable encryption based on sliding window (SCSW) and the traditional Feistel architecture [14]. Besides, it significantly cut the time complexity. Moreover, their algorithm can be installed in some hardware platform.

An other work [26] secured the Internet of Things (IoT) communication network. This work introduced a dynamic encryption algorithm (DSIT) for the security of the IoT communication. The suggested algorithm employed the Feistel and Substitution–Permutation Network. Apart from that, DSIT being a block cipher, takes input of the 64-bit block of plaintext, 64-bit secret key, and a secret dynamic box (D-box). Resultantly, it renders a 64-bit ciphertext after carrying out the 8 rounds of suggested algorithm of DSIT. During each round, both D-box and key are upgraded. The incorporation of this dynamic handling renders heightened security to the corresponding IoT network. Raspberry Pi 3

Model B+ was used as a platform and Avalanche effect to the tune of 50% average was obtained. Lastly, the security analyses validated that the suggested algorithm is furnished with a capability of defying the varied cryptanalytic attacks.

By employing the theory of chaos, the work [44] wrote a lightweight nonlinear mechanism to secure the digital images in the global networks. Moreover, a block cipher through the usage of theory of lightweight chaos, was implemented. More than one chaotic systems were exploited to spawn the random keys for each channel of the given image. In particular, chaotic gingerbread map and Arnold cat map were utilized for the realization of diffusion and confusion effects. In the confusion phase, the given pixels were re-ordered. In the same fashion, image pixels were disturbed in the diffusion phase through the aid of gingerbread map. Varied security parameters were used to demonstrate the prowess and immunity of the suggested lightweight image cipher from the attacks.

Table 1 summarizes representative recent image encryption schemes in terms of their principal methodology, key space, and commonly reported security measures. Although existing approaches demonstrate competitive statistical and differential security, many employ conventional row-, column-, or pixel-level permutation mechanisms, while some introduce comparatively complex transformation procedures. Furthermore, lightweight operation remains particularly important for resource-constrained applications. These observations motivate the proposed approach, which introduces chaos-controlled cross-channel diagonal-strip permutation combined with computationally simple XOR diffusion. The objective is to provide effective confusion and diffusion while retaining relatively simple primitive operations suitable for resource-conscious image security applications.

By taking inspiration of the above literature, this work is providing a solution to the hot problem of lightweight cryptography of the images. To spawn the streams of random numbers, this work has used the Intertwining Logistic Chaotic map [43]. Moreover, a novel idea of scrambling the pixels data of the given color image has been conceived in this study. As the given color image is input, it is decomposed into its principal planes of red, green and blue. Two randomly chosen diagonal strips from any two arbitrary planes have been swapped a large number of times to inject the necessary feature of confusion/scrambling in the given plain image. Chaotic data governs the selection of two planes out of the three planes in each iteration. To increase the security effects, XoR operation has also been carried out between the confused image and the stream of random numbers.

Motivated by the aforementioned challenges, this study aims to design, implement, and evaluate a lightweight chaos-based encryption scheme for RGB images, with particular emphasis on computational simplicity and security for resource-conscious applications. The proposed framework investigates whether chaos-controlled cross-channel diagonal-strip permutation, combined with lightweight

XOR-based diffusion, can provide effective image protection without introducing computationally expensive operations. Accordingly, the following research questions are formulated:

- RQ1:** Can chaos-controlled diagonal-strip permutation across RGB components effectively reduce the statistical and spatial dependencies present in plaintext images?
- RQ2:** Can the integration of Intertwining Logistic Map (ILM)-based chaotic sequences with XOR diffusion provide adequate resistance against statistical, differential, and key-related attacks?
- RQ3:** Can the proposed encryption framework achieve competitive computational performance while maintaining a lightweight algorithmic structure suitable for resource-conscious IoT image security applications?

Based on these research questions, the principal objectives of this study are as follows:

- To design a lightweight RGB image encryption scheme based on ILM-controlled dynamic diagonal-strip permutation and XOR-based diffusion.
- To evaluate the security of the proposed scheme using established statistical, differential, key-sensitivity, cryptanalytic, and reconstruction measures.
- To assess the computational efficiency of the proposed scheme through encryption time, throughput, and theoretical computational and memory complexity analyses.

The intended outcome of this study is a reversible and computationally simple RGB image encryption framework that provides strong image randomization and resistance to commonly considered cryptanalytic attacks, while demonstrating the potential for resource-conscious IoT applications. Direct implementation and validation on resource-constrained embedded hardware remain subjects for future investigation.

Having said that, the following bullet points shed light on the salient features of the suggested work:

- Security has been imparted to the RGB images for the IoT devices.
- Randomly generated diagonal strips of varied lengths have been swapped with each other for a number of times to inject the confusion effects. Previous studies worked only on the horizontal and vertical strips. In this way, the proposed approach boosted the security effects.
- Comprehensive security analysis has been carried out to validate the idea given in the second bullet.

Table 1: Overview of representative state-of-the-art image encryption schemes

Scheme	Main Technique	Key Space	Entropy	NPCR (%)	UACI (%)	Time (s)
Ref. [24]	Chaotic map + Elliptic curve	2^{511}	7.99	99.58	33.18	0.5634
Ref. [6]	Multiple chaotic maps	10^{144}	7.9972	99.6254	33.0704	2.750966
Ref. [8]	Hyperchaotic system	10^{3168}	7.99911	99.6078	29.4973	0.230717
Ref. [27]	Chaotic map + self reversible strategy	2^{512}	7.902474	99.61306	33.46204	0.9196
Ref. [30]	Chaotic map framework	2^{568}	7.9934	99.607	33.44	-

Remaining work has been scheduled as follows. Section 2 describes the chaotic systems and their importance. Moreover, this section also mentions the Intertwining Logistic Map which has been used in the proposed work. In the Section 3, suggested algorithm has been narrated in detail. Moreover, the simulation has been carried out in the Section 4 to demonstrate the workability of the suggested encryption and decryption algorithms. To validate the proposed work, Section 5 evaluates the suggested algorithm by using the various state of the art security parameters. Lastly, the Section 7 closes the paper by providing the concluding remarks.

2 Chaotic systems

Chaotic systems are dynamic and highly sensitive systems in which small variations in initial conditions can lead to significantly different outcomes over time [41]. This sensitivity to initial conditions is a fundamental aspect of chaos theory, often illustrated by the “butterfly effect.” The butterfly effect suggests that the flap of a butterfly’s wings in one part of the world could set off a chain of events that might eventually lead to a tornado or a typhoon in another part of the world. In chaotic systems, even tiny disturbances can amplify and lead to unpredictable and divergent trajectories, making long-term predictions challenging or impossible. This work has employed Intertwining Logistic Map [20].

$$\begin{cases}
 x(n+1) = [\mu \times k_1 \times y(n) \times (1 - x(n) + z(n))] \bmod 1 \\
 y(n+1) = [\mu \times k_2 \times y(n) + z(n) \times 1/(1+x^2(n+1))] \bmod 1 \\
 z(n+1) = [\mu \times (x(n+1) + y(n+1) + k_3) \times \sin z_n] \bmod 1
 \end{cases} \quad (1)$$

The generation of chaotic data is subject to some conditions. These conditions include $0 < \mu \leq 3.999$, $|k_1| > 33.5$, $|k_2| > 37.9$, $|k_3| > 35.7$. Moreover, $n = 0, 1, 2, 3, \dots$. This 3D map exhibits more pronounced chaotic dynamics compared to the previous logistic map. Additionally, this map displays a uniform distribution (as seen in Figures 1a to 1c) and does not exhibit any vacant regions [20]. Besides, Lyapunov exponent of ILM is positive (Figure 1d) which is symptomatic to the fact that it is furnished with nicer behavior of chaoticity.

3 Suggested lightweight RGB image cipher

The proposed encryption scheme is designed with an emphasis on computational simplicity for resource-conscious image security applications. Its lightweight nature primarily arises from the use of simple diagonal-strip permutation and bitwise XOR-based diffusion operations, thereby avoiding computationally intensive transformations during the main encryption stages. In the confusion stage, dynamically selected diagonal strips are exchanged between selected RGB components to disturb both the spatial arrangement and cross-channel distribution of image pixels. The diffusion stage subsequently employs simple bitwise XOR operations to modify pixel values with comparatively low computational overhead. Chaotic sequences are utilized to control the dynamic operations required by these stages. These design choices are intended to limit computational complexity while maintaining the confusion and diffusion properties required for image encryption, making the proposed framework a potential candidate for resource-constrained IoT applications, subject to further validation on embedded IoT hardware.

Figure 2 shows suggested algorithm consisting of some phases. In first phase, a color image *img* has been given to the proposed algorithm with the size of $m \times n \times 3$. For embedding plaintext sensitivity to the suggested cipher, an image characteristic using the formula $sum(sum(sum(img)))$ has been extracted from the given RGB image. We describe the peculiar way of this formula at the appropriate place. The incorporation of plaintext sensitivity defies the potential differential attacks launched by the hackers and the other adversaries. For all the given input plain images, distinct random numbers streams would be generated—the underlying philosophy behind the notion of plaintext sensitivity [36]. In the second phase, the Intertwining Logistic Map (ILM) has been ignited by giving it security key $x_0, y_0, z_0, \mu, k_1, k_2$, and k_3 . This map rendered the streams *choice, direction, x, y, mask* of random numbers. In the third phase, confusion, which is one of the chief operations of any cryptographic product, has been carried out through the module of *Diagonal Strips Based Scrambler* (DSBS). In the fourth phase, mask image *mask* has been generated through the module of Key Image formation. Lastly, to add effects of diffusion in the proposed image cipher, diffusion operation was made between the scrambled image and key/mask image to get the final color cipher image.

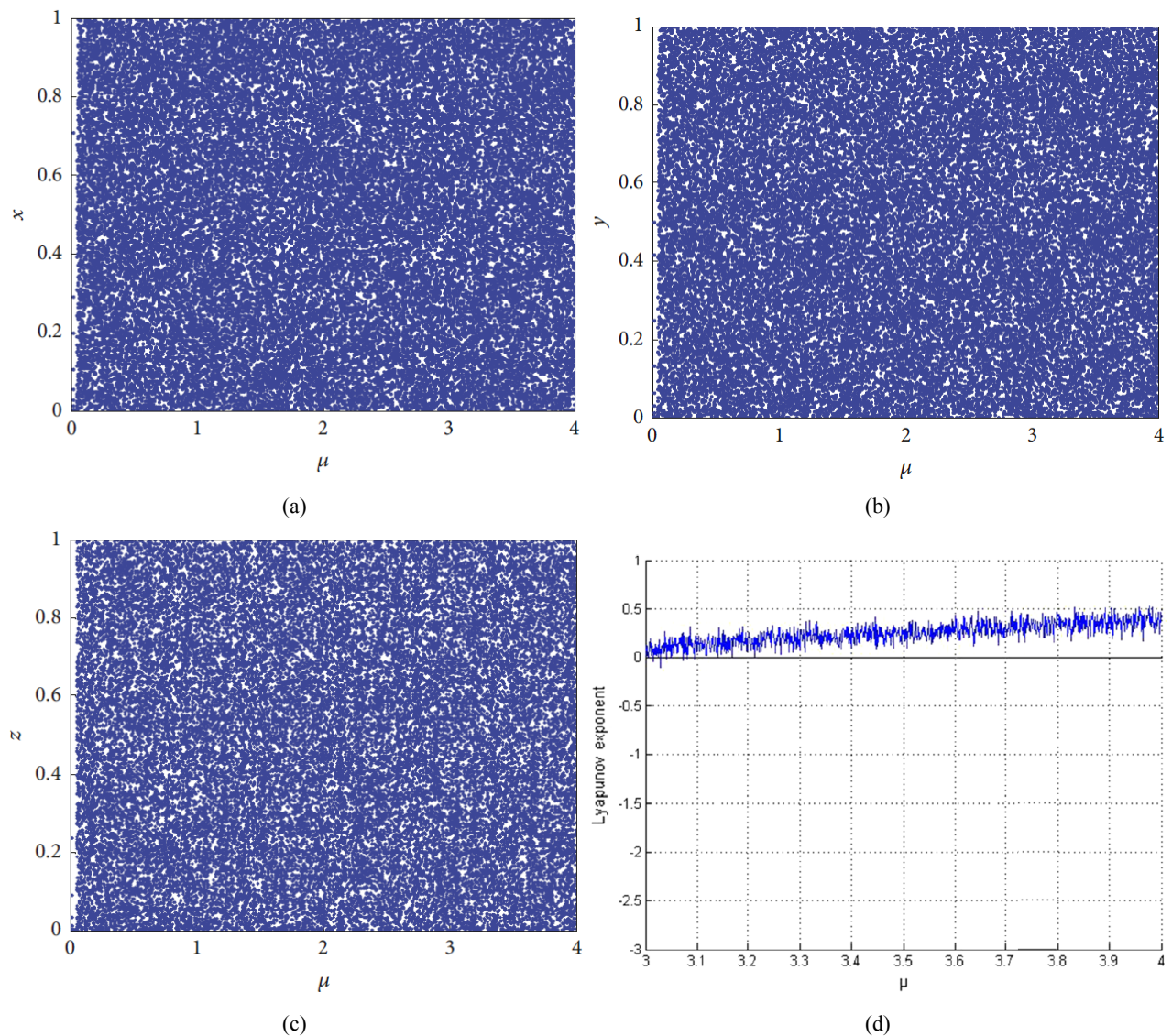


Figure 1: 3D chaotic intertwining logistic map with the distribution of its streams: (a) sequence x bifurcation diagram; (b) sequence y bifurcation diagram; (c) sequence z bifurcation diagram; (d) lyapunov exponent

3.1 Development of initial values of chaotic map/system

In this study, this is our assumption that $m \times n \times 3$ are the dimensions/size of the given input color image img . Call the Algorithm (1) for developing streams of random numbers with the tuple of parameters $x_0, y_0, z_0, \mu, k_1, k_2, k_3, m, n$.

Here, this algorithm will be explained in detail. Line 1 finds sum of all pixels for the three planes of given image img . Since image img under consideration is 3-dimensional, hence the built-in function *sum* has been invoked for three times to get the final sum of all the pixels. Moreover, it divides the result with 2^{32} . In this way, the value of x_0 will stay in the allowed range. In the line 2, System (1) has been invoked which returns these random numbers streams, i.e., $\{x(t)\}_{t=1}^{3mn}$,

$\{y(t)\}_{t=1}^{dmn}$, $\{z(t)\}_{t=1}^{3mn}$. Lines 3 —7 initialize the streams $\{choice(t)\}_{t=1}^{3mn}$, $\{direction(t)\}_{t=1}^{3mn}$, $\{p(t)\}_{t=1}^{3mn}$, $\{q(t)\}_{t=1}^{3mn}$, $\{mask(t)\}_{t=1}^{3mn}$ with the arrays of zeros having the dimensions of $(1, 3 \times m \times n)$. Next, the *for* loop (Line 8) iterates for $3mn$ times. We get the required streams $\{choice(t)\}_{t=1}^{3mn}$, $\{direction(t)\}_{t=1}^{3mn}$, $\{p(t)\}_{t=1}^{3mn}$, $\{q(t)\}_{t=1}^{3mn}$, $\{mask(t)\}_{t=1}^{3mn}$ of random numbers in the ranges $[1, 3]$, $[1, 2]$, $[1, 256]$, $[1, 256]$, $[0, 255]$ respectively. Lastly, this algorithm returns the streams $choice, direction, p, q, mask$.

3.2 Image encryption procedure

Algorithm (2) named as *Diagonal Strips Based Scrambler* (DSBS) has been called with the parameters of $img, choice, direction, p, q, m, n$. Here we explain this Algorithm in different steps.

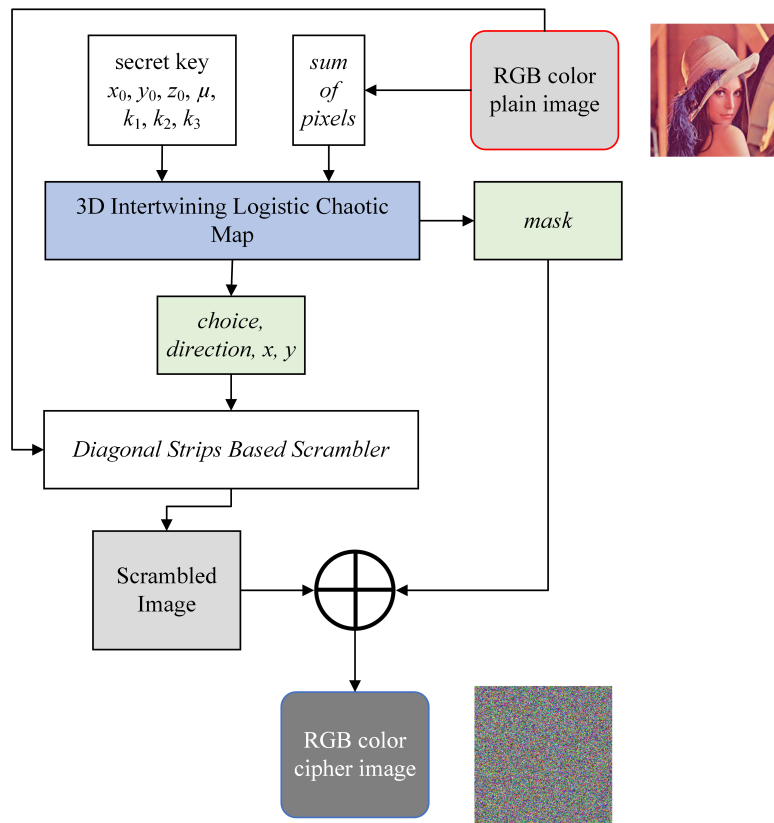


Figure 2: Proposed lightweight image cipher based on dsbs

Algorithm 1: Generation of chaotic data**Input:** $img, x_0, y_0, z_0, \mu, k_1, k_2, k_3, m, n$ **Output:** $choice, direction, p, q, mask$

- 1: $x_0 \leftarrow x_0 + \frac{sum(sum(sum(img)))}{2^{32}}$
- 2: Spark the System (1) with the parameters $x_0, y_0, z_0, \mu, k_1, k_2, k_3, m, n$ which returns the streams $\{x(t)\}_{t=1}^{3mn}, \{y(t)\}_{t=1}^{dmn}, \{z(t)\}_{t=1}^{3mn}$.
- 3: $\{choice(t)\}_{t=1}^{3mn} \leftarrow zeros(1, 3 \times m \times n)$
- 4: $\{direction(t)\}_{t=1}^{3mn} \leftarrow zeros(1, 3 \times m \times n)$
- 5: $\{p(t)\}_{t=1}^{3mn} \leftarrow zeros(1, 3 \times m \times n)$
- 6: $\{q(t)\}_{t=1}^{3mn} \leftarrow zeros(1, 3 \times m \times n)$
- 7: $\{mask(t)\}_{t=1}^{3mn} \leftarrow zeros(1, 3 \times m \times n)$
- 8: **for** $t \leftarrow 1$ to $3mn$ **do**
- 9: $choice(t) \leftarrow \text{mod}(\text{floor}(x(t) \times 10^{14}), 3) + 1$
- 10: $direction(t) \leftarrow \text{mod}(\text{floor}(y(t) \times 10^{14}), 2) + 1$
- 11: $p(t) \leftarrow \text{mod}(\text{floor}(z(t) \times 10^{14}), 256) + 1$
- 12: $q(t) \leftarrow \text{mod}(\text{floor}(x(t) \times y(t) \times 10^{14}), 256) + 1$
- 13: $mask(t) \leftarrow \text{mod}(\text{floor}((x(t) + y(t)) \times 10^{14}), 256)$
- 14: **end for**
- 15: **return** $choice, direction, p, q, mask$

Step 1: Lines (1-3) access three constituent planes of *red, green* and *blue* from the given color image *img*.

Step 2: Line 4 uses the *for* loop which iterates for $3mn$ times.

Step 3: The *switch – case* structure (Lines 5-12) uses the index $choice(i)$. In each case, the Algorithm 3 *SwapDiagStrips* has been invoked with the parameters of

two planes (out of the three planes *red, green* and *blue*), $direction(i)$, $p(i)$, $q(i)$, m and n . This algorithm returns those two planes after swapping the diagonal strips. We explain this algorithm shortly.

Step 4: Line 14 initializes the array img' with the values of zeros having the dimensions of $(m, n, 3)$.

Step 5: Lines (15-17) merge the constituent planes to come

up with the final scrambled image img' and this image has been returned by the last line 18.

Now, we will explain the Algorithm 3 which carries out the main “manufacturing” task of the proposed scheme. This algorithm fundamentally consists of *if* part (Lines 3-19) and *else* part (Lines 21-36).

Step 6: Lines (1 - 2) save the current values of p and q to the variables $savep$ and $saveq$ respectively.

Step 7: If the condition at the line 3 is true, then the *while* loop at the line 4 iterates to swap the pixels of the diagonal strips of the two images img_1 and img_2 .

Step 8: Lines 8 and 9 update the values of p and q so that the coming pixels of the diagonal strips may be accessed.

Step 9: Lines 11 and 12 initialize the values of p and q for the other part of the diagonal strip.

Step 10: The *while* loop at the line 13 swaps the remaining pixels of the diagonal strips.

Step 11: In the same way, if the *if* condition at the line 3 evaluates to be false, the swapping of the pixels between the diagonal strips of the two images img_1 and img_2 has been carried out in the *else* part spanning the lines (21 - 36). Finally, lines 38 and 39 assign to the images img'_1 , img'_2 the confused images.

Step 12: Line 40 returns the confused images img'_1 and img'_2 to the calling program.

Step 13: Resize mask image $mask$ as $(m, n, 3)$. To embed the necessary diffusion effects, the following XoR operation between mask image $mask$ and scrambled image img' has been performed.

$$img''(i) = mask(i) \oplus img'(i) \quad (2)$$

where $1 \leq i \leq 3mn$. img'' is the final color cipher image.

Unlike conventional permutation mechanisms that primarily rearrange pixels along rows, columns, or predetermined spatial positions, the proposed method performs cross-channel permutation through dynamically selected diagonal strips. The diagonal-strip swapping operation simultaneously disturbs the spatial arrangement of pixels and their distribution across different RGB components. Furthermore, the dynamic selection of the participating color components and diagonal regions introduces variability into the confusion process. This design, combined with lightweight bitwise diffusion, aims to achieve effective image randomization without introducing computationally expensive operations.

3.3 Image decryption procedure

The decryption process is carried out by reversing the operations performed during encryption. Since the XOR operation is self-inverse, the diffusion stage can be reversed by applying the same mask image to the cipher image. Let img'' denote the cipher image and $mask$ denote the identical mask regenerated using the secret key. The scrambled image img' is recovered as

$$img'(i) = mask(i) \oplus img''(i), \quad (3)$$

where $1 \leq i \leq 3mn$. This operation successfully removes the diffusion introduced during encryption because

$$(img'(i) \oplus mask(i)) \oplus mask(i) = img'(i). \quad (4)$$

After recovering the scrambled image, the confusion stage is reversed by applying the diagonal-strip swapping operations in the reverse order. Specifically, whereas the encryption process performs the swapping operations from $i = 1$ to $3mn$, the decryption process (Algorithms 4 and 5) applies the corresponding operations from $i = 3mn$ down to 1. Since each pixel-swapping operation is self-inverse, reapplying the same swap to the same pair of image components restores their previous values. Consequently, the combined inverse XOR and inverse diagonal-strip operations reconstruct the original red, green, and blue components of the plaintext image.

Since the proposed cipher is based on the principles of private key cryptography, so, secret key will be shared manually to the destination.

4 Simulation and machine experimentation

To evaluate the effectiveness of suggested image security algorithm, we have chosen four color images whose names are Lena, Baboon, House, and Starfish, each with a size of 256×256 pixels. If someone is interested in these images, they can be downloaded from the USC-SIPI Image Database. Experimentation process was conducted using MATLAB R2024a version.

To generate the random numbers, system's parameters and initial values of the used chaotic map (3D Intertwining Logistic Map) were set as follows: $x_0 = 0.36$, $y_0 = 0.25$, $z_0 = 0.78$, $\mu = 1.5$, $k_1 = 35.5$, $k_2 = 38.2$, and $k_3 = 36.0$ to initiate System (1). The Figures 3, 4, and 5 depict plaintext input images, ciphertext output images, and recovered images, respectively. It is noteworthy that the plaintext images have been transformed into completely unrecognizable forms, confirming the successful encryption process. Additionally, these encrypted images have been successfully decrypted, demonstrating effectiveness of the decryption process proposed in this work.

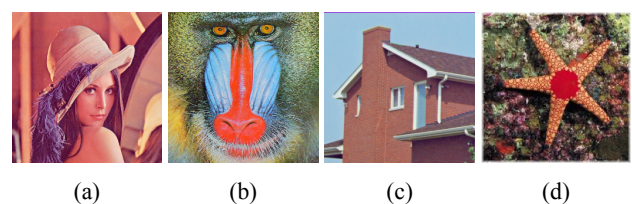


Figure 3: Plaintext images: (a) lena; (b); baboon; (c) house; (d) starfish

Algorithm 2: Diagonal strips based scrambler (dsbs)**Input:** *img, choice, direction, p, q, m, n***Output:** *img'*

```

1: red  $\leftarrow$  img(:,: ,1)
2: green  $\leftarrow$  img(:,: ,2)
3: blue  $\leftarrow$  img(:,: ,3)
4: for i  $\leftarrow$  1 to 3mn do
5:   switch (choice(i))
6:   case 1:
7:     [red, green]  $\leftarrow$  SwapDiagStrips(red, green, direction(i), p(i), q(i), m, n)
8:   case 2:
9:     [green, blue]  $\leftarrow$  SwapDiagStrips(green, blue, direction(i), p(i), q(i), m, n)
10:  case 3:
11:    [red, blue]  $\leftarrow$  SwapDiagStrips(red, blue, direction(i), p(i), q(i), m, n)
12:  end switch
13: end for
14: img'  $\leftarrow$  zeros(m, n, 3)
15: img'(:,: ,1)  $\leftarrow$  red
16: img'(:,: ,2)  $\leftarrow$  green
17: img'(:,: ,3)  $\leftarrow$  blue
18: return img'

```

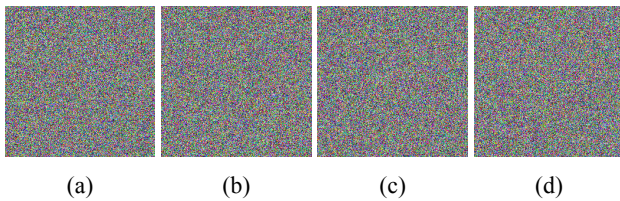


Figure 4: The ciphertext output images:(a) lena; (b) ba-boon; (c) house; (d) starfish

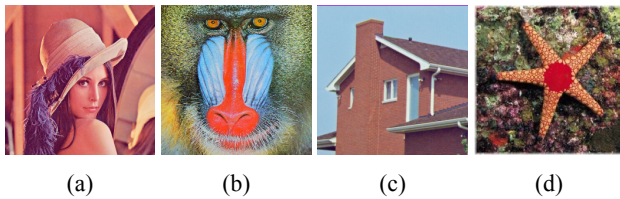


Figure 5: The decrypted/retrieved images: (a) lena; (b) ba-boon; (c) house; (d) starfish

5 Security and performance analyses with varied metrics

To demonstrate resilience of suggested image cryptosystem against various attacks from hacker community, a wide range of validation parameters/metrics have been devised. In upcoming subsections, we would employ some of the most widely recognized ones.

5.1 Analysis of key space

A brute-force assault is a tactic employed by adversaries where they systematically test all possible secret keys in

order to discover the correct one. This approach necessitates a substantial key space. The cryptographers recommend setting the minimum key space of 2^{100} [9]. In this research, system parameters and initial values function as the keys for the image cipher. With 7 such key components, the key space expands to $10^{14 \times 7} = 10^{98}$, where number 10^{-14} is acting as computer precision of the system where this project has been carried out. This magnitude offers robust protection against brute-force assault. Furthermore, Table 2 provides a key space comparison between suggested cipher and other image security solutions within the field. Unfortunately, we could not beat any of the studies but we have maintained the minimum deterrence of 2^{100} .

5.2 Key sensitivity

Key sensitivity is an important security property of an image cryptosystem, as a very small modification in the secret key should produce a substantially different cipher image. A cipher exhibiting strong key sensitivity reduces the possibility of obtaining meaningful information using a key that differs only slightly from the correct one.

Let the original secret key be represented as

$$SK_0 = \{x_0, y_0, z_0, \mu, k_1, k_2, k_3\}. \quad (5)$$

To evaluate the sensitivity of the proposed scheme, a small perturbation $\delta = 10^{-14}$ is introduced into one key parameter while all other parameters remain unchanged. For example, perturbing the initial condition x_0 gives

$$x'_0 = x_0 + \delta, \quad (6)$$

which results in the modified key

Algorithm 3: Diagonal-strip swapping**Input:** $img_1, img_2, direction, p, q, m, n$ **Output:** img'_1, img'_2

```

1:  $savep \leftarrow p$ 
2:  $saveq \leftarrow q$ 
3: if  $direction = 1$  then
4:   while  $p \geq 1 \ \&\& \ q \leq m$  do
5:      $temp \leftarrow img_1(p, q)$ 
6:      $img_1(p, q) \leftarrow img_2(p, q)$ 
7:      $img_2(p, q) \leftarrow temp$ 
8:      $p \leftarrow q - 1$ 
9:      $q \leftarrow q + 1$ 
10:  end while
11:   $p \leftarrow savep + 1$ 
12:   $q \leftarrow saveq - 1$ 
13:  while  $p \leq m \ \&\& \ q \geq 1$  do
14:     $temp \leftarrow img_1(p, q)$ 
15:     $img_1(p, q) \leftarrow img_2(p, q)$ 
16:     $img_2(p, q) \leftarrow temp$ 
17:     $p \leftarrow p + 1$ 
18:     $q \leftarrow q - 1$ 
19:  end while
20: else
21:  while  $p \geq 1 \ \&\& \ q \geq 1$  do
22:     $temp \leftarrow img_1(p, q)$ 
23:     $img_1(p, q) \leftarrow img_2(p, q)$ 
24:     $img_2(p, q) \leftarrow temp$ 
25:     $p \leftarrow p - 1$ 
26:     $q \leftarrow q - 1$ 
27:  end while
28:   $p \leftarrow savep + 1$ 
29:   $q \leftarrow saveq + 1$ 
30:  while  $p \leq m \ \&\& \ q \leq n$  do
31:     $temp \leftarrow img_1(p, q)$ 
32:     $img_1(p, q) \leftarrow img_2(p, q)$ 
33:     $img_2(p, q) \leftarrow temp$ 
34:     $p \leftarrow p + 1$ 
35:     $q \leftarrow q + 1$ 
36:  end while
37: end if
38:  $img'_1 \leftarrow img_1$ 
39:  $img'_2 \leftarrow img_2$ 
40: return  $img'_1, img'_2$ 

```

$$SK_1 = \{x'_0, y_0, z_0, \mu, k_1, k_2, k_3\}. \quad (7)$$

The same procedure is independently applied to the remaining key parameters to examine their influence on the encryption output. The resulting modified keys are denoted by SK_t , where $t = 1, 2, \dots, 14$. For each experiment, the same plaintext image is encrypted separately using the original key SK_0 and the corresponding perturbed key SK_t . The percentage difference between the resulting cipher images is then calculated to quantify the effect of the small key perturbation.

Table 3 summarizes the key-sensitivity results. The minimum observed difference between two cipher images is 99.5876%, indicating that even a perturbation of only 10^{-14} in a key parameter produces a substantially different encryption output. Furthermore, the proposed scheme achieves an average key sensitivity value of approximately 99.61%, which is better than the result reported in [40]. These results demonstrate the strong sensitivity of the proposed encryption scheme to small variations in its secret key parameters, thereby strengthening its resistance to attacks that rely on approximating or slightly modifying the

Algorithm 4: Inverse diagonal strips based scrambler (i-dsbs)**Input:** img' , $choice$, $direction$, p , q , m , n **Output:** img

```

1:  $red \leftarrow img'(:, :, 1)$ 
2:  $green \leftarrow img'(:, :, 2)$ 
3:  $blue \leftarrow img'(:, :, 3)$ 
4: for  $i \leftarrow 3mn$  downto 1 do
5:   if  $choice(i) = 1$  then
6:      $[red, green] \leftarrow InverseSwapDiagStrips(red, green, direction(i), p(i), q(i), m, n)$ 
7:   else if  $choice(i) = 2$  then
8:      $[green, blue] \leftarrow InverseSwapDiagStrips(green, blue, direction(i), p(i), q(i), m, n)$ 
9:   else
10:     $[red, blue] \leftarrow InverseSwapDiagStrips(red, blue, direction(i), p(i), q(i), m, n)$ 
11:   end if
12: end for
13:  $img \leftarrow zeros(m, n, 3)$ 
14:  $img(:, :, 1) \leftarrow red$ 
15:  $img(:, :, 2) \leftarrow green$ 
16:  $img(:, :, 3) \leftarrow blue$ 
17: return  $img$ 

```

correct key.

5.3 Statistical analysis

Statistical analysis encompasses gathering, interpreting, and presenting data to reveal patterns, trends, and relationships, facilitating well-informed decision-making. It employs various mathematical and computational methods to analyze and draw meaningful conclusions from data sets.

A statistical attack is a commonly employed strategy by hackers against ciphers. For showing the resilience of suggested image crypto-system against such attacks, this study utilizes two specific measures: namely, the examination of histogram and the and correlation analyses.

5.3.1 Histogram

Distribution of pixel color values is denoted by a graphical tool known as a histogram. Typically, the histogram of plaintext images exhibits a pronounced slanting pattern, characterized by peaks and valleys. Such histograms capture the interest of cryptanalysts due to their information-rich nature, making them susceptible to histogram-based attacks. However, when an image is encoded by employing encryption technique over it, resulting histogram of ciphered image shows a consistently smooth and uniform distribution, which is highly resistant to attacks of histogram [29].

Histogram analyses for ciphertext and plaintext images of Lena are presented in Figures 6 and 7, in a respective way. The figures vividly illustrate that the plain image exhibits a dynamic and slanting histogram pattern, whereas encrypted image histogram is notably uniform and smooth. This uniformity and smoothness serve as robust defenses

against potential histogram-based attacks, reinforcing security and resilience of suggested approach.

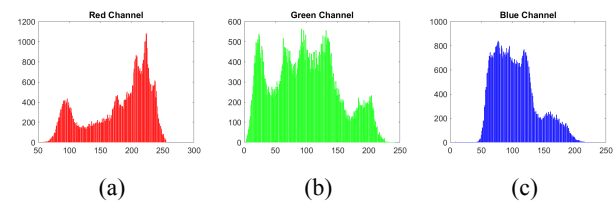


Figure 6: Plaintext lena image histogram in varied components. (a) red, (b) green, (c) blue

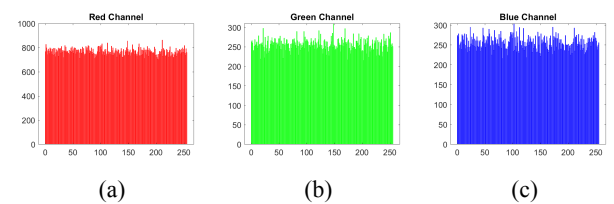


Figure 7: Ciphertext lena image histogram in varied components. (a) red, (b) green, (c) blue

Moreover, scientists and cryptographers employ the concept of variance to quantitatively assess the smoothness of a histogram. Lower values of variance indicate a smoother histogram, and conversely [3]. Table 4 displays the variance values of histogram distributions for ciphered images of Lena, Baboon, House and Starfish. Based on the table, the calculated average variance value for the selected images is 266.0039. Additionally, our research surpasses the findings in the study by [47]. Consequently, we argue that our proposed approach is both effective and secure.

Algorithm 5: Inverse diagonal-strip swapping**Input:** $img'_1, img'_2, direction, p, q, m, n$ **Output:** img_1, img_2

```

1:  $savep \leftarrow p$ 
2:  $saveq \leftarrow q$ 
3: if  $direction = 1$  then
4:   while  $p \geq 1 \ \&\& \ q \leq m$  do
5:      $temp \leftarrow img'_1(p, q)$ 
6:      $img'_1(p, q) \leftarrow img'_2(p, q)$ 
7:      $img'_2(p, q) \leftarrow temp$ 
8:      $p \leftarrow q - 1$ 
9:      $q \leftarrow q + 1$ 
10:  end while
11:   $p \leftarrow savep + 1$ 
12:   $q \leftarrow saveq - 1$ 
13:  while  $p \leq m \ \&\& \ q \geq 1$  do
14:     $temp \leftarrow img'_1(p, q)$ 
15:     $img'_1(p, q) \leftarrow img'_2(p, q)$ 
16:     $img'_2(p, q) \leftarrow temp$ 
17:     $p \leftarrow p + 1$ 
18:     $q \leftarrow q - 1$ 
19:  end while
20: else
21:  while  $p \geq 1 \ \&\& \ q \geq 1$  do
22:     $temp \leftarrow img'_1(p, q)$ 
23:     $img'_1(p, q) \leftarrow img'_2(p, q)$ 
24:     $img'_2(p, q) \leftarrow temp$ 
25:     $p \leftarrow p - 1$ 
26:     $q \leftarrow q - 1$ 
27:  end while
28:   $p \leftarrow savep + 1$ 
29:   $q \leftarrow saveq + 1$ 
30:  while  $p \leq m \ \&\& \ q \leq n$  do
31:     $temp \leftarrow img'_1(p, q)$ 
32:     $img'_1(p, q) \leftarrow img'_2(p, q)$ 
33:     $img'_2(p, q) \leftarrow temp$ 
34:     $p \leftarrow p + 1$ 
35:     $q \leftarrow q + 1$ 
36:  end while
37: end if
38:  $img_1 \leftarrow img'_1$ 
39:  $img_2 \leftarrow img'_2$ 
40: return  $img_1, img_2$ 

```

5.3.2 Correlation coefficient analysis

The correlation coefficient measures both the extent and direction of a linear relationship between two variables, with values spanning from -1 (indicating perfect negative correlation) to 1 (indicating perfect positive correlation). In normal and unaltered images, pixels exhibit a high degree of correlation, leading to elevated correlation coefficient values. However, when a plain image undergoes an image encryption process, this robust correlation diminishes, resulting in a substantial drop in the coefficient value. The correlation coefficient typically ranges from -1 to 1 as described

earlier, where a value of zero indicates no discernible correlation among adjacent pixels, while values of 1 or -1 signify maximum pixel correlation.

To illustrate this characteristic of the proposed scheme, 5,000 randomly selected pairs of pixels were analyzed in both ciphered and plain images. This analysis was conducted in diagonal, vertical, and horizontal directions, employing the following formula for correlation coefficient (CC) [19]:

Table 2: Key space of suggested technique and its comparison with other schemes

Study	Key space
Suggested	$10^{98} \approx 2^{325}$
Ref. [45]	2^{425}
Ref. [17]	10^{144}
Ref. [46]	2^{497}
Ref. [7]	10^{144}
Ref. [5]	2^{1754}

$$CC = \frac{N \sum_{j=1}^N (x_j \times y_j) - \sum_{j=1}^N x_j \times \sum_{j=1}^N y_j}{\sqrt{\left(N \sum_{j=1}^N x_j^2 - \left(\sum_{j=1}^N x_j \right)^2 \right) \left(N \sum_{j=1}^N y_j^2 - \left(\sum_{j=1}^N y_j \right)^2 \right)}} \quad (8)$$

The equation pertains to the image's total pixel count denoted by the variable N , with consecutive pixel color values denoted as x and y . Figure 8 depicts the correlation distribution among neighboring pixels in diagonal, horizontal, and vertical directions for both the original Lena image and its encrypted counterpart. Table 5 depicts this coefficient be-

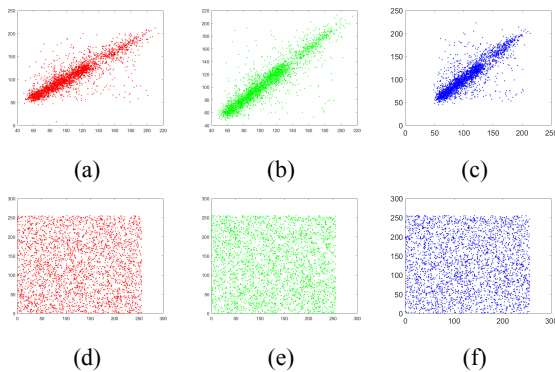


Figure 8: Distribution of pixel correlations in the specified channel and lena image: (a) red component, horizontal, original image; (b) green component, vertical, original image; (c) blue component, diagonal, original image; (d) red component, horizontal, cipher image; (e) green component, vertical, cipher image; (f) blue component, diagonal, cipher image

tween two adjacent pixels for cipher and plain images of Lena. Based on the data provided in this table, the metric value is nearly equal to 1 for the original image, whereas it approaches zero for encrypted/ciphered images. Figure 8 and Table 5 collectively emphasize the significant reduction in the potential correlation between pixels in ciphered and plain images. Furthermore, to underscore the superiority of the proposed approach over other published works, a comparative analysis is presented in Table 6. Undoubtedly,

the results of suggested work are highly competitive. It is to be noted that these metrics having been obtained by taking the average values of three color planes of red, green and blue.

5.4 Entropy analysis

Entropy or Information entropy (IE) characterizes degree of randomness and disorder present in given image pixels. When assessing the security of an encrypted image, a higher information entropy value is desirable. Essentially, this metric quantifies the extent to which the pixels have been rearranged, providing a measure of the signal's unpredictability and randomness. In 1949, Claude Shannon introduced a formula to quantify this notion, as documented in his work [25].

$$E(t) = \sum_{k=0}^{2^z-1} p(t_k) \log \frac{1}{p(t_k)} \quad (9)$$

In the above mathematical formula, $E(t)$ denotes an information source's entropy denoted as t . Additionally, $p(t_k)$ signifies the notion of probability of given information source t_k . When an image has been ideally randomized with 256 different grayscale values, peak attainable value for entropy comes out to be 8. To ensure strong security in image ciphers, it is desirable for this metric to approach the value of 8.

The table in reference 7 displays the information entropy values for various images. It is obvious that average information entropy values across the four selected RGB images closely approximate the ideal value of 8. Hence, our scheme demonstrates resilience against entropy-based attacks. A comparison of our proposed scheme with other schemes is also detailed in Table 7. Notably, the information entropy value for the Lena RGB image in our proposed scheme surpasses that of the references [45, 7, 22] for the 256×256 - sized images. Besides, suggested work beats the works [5, 18] in case image size is 512×512 .

5.5 Avalanche attack/plaintext sensitivity analysis

Assessment of the attack's resilience relies on two security parameters namely, Unified Average Changing Intensity (UACI), the Number of Pixels Change Rate (NPCR) as described by [49]. These parameters evaluate the impact of modifying a one pixel value on cipher image. In this particular analysis, attackers introduce a subtle change in given plaintext image. Subsequently, they create two copies/versions of cipher image: one using unaltered plaintext image and other using the plaintext image with a little modification. This process is the essence of the special jargon "Differential." By carefully examining the potential relation between these two encrypted/coded images of plaintext, they aim to uncover potential patterns. Through

Table 3: Difference rates between cipher images generated using slightly different keys.

Security keys (SK)	Rates of difference(%)			
	Lena	Baboon	House	Star Fish
$SK_1(x_0' = x_0 + \delta)$	99.6170	99.6287	99.6209	99.5995
$SK_2(y_0' = y_0 + \delta)$	99.6271	99.6058	99.6312	99.6006
$SK_3(z_0' = z_0 + \delta)$	99.6166	99.6046	99.6276	99.6112
$SK_4(\mu = \mu + \delta)$	99.6373	99.6023	99.6299	99.6051
$SK_5(k_1' = k_1 + \delta)$	99.6169	99.5876	99.6256	99.6376
$SK_6(k_2' = k_2 + \delta)$	99.6267	99.6265	99.6109	99.6256
$SK_7(k_3' = k_3 + \delta)$	99.6151	99.6111	99.6276	99.6055
$SK_8(x_0' = x_0 - \delta)$	99.6363	99.6098	99.6213	99.6089
$SK_9(y_0' = y_0 - \delta)$	99.6150	99.6372	99.6266	99.6273
$SK_{10}(z_0' = z_0 - \delta)$	99.6168	99.6261	99.6209	99.6361
$SK_{11}(\mu = \mu - \delta)$	99.6374	99.5987	99.6156	99.6100
$SK_{12}(k_1' = k_1 - \delta)$	99.6312	99.6209	99.6122	99.5876
$SK_{13}(k_2' = k_2 - \delta)$	99.6199	99.6097	99.5906	99.6069
$SK_{14}(k_3' = k_3 - \delta)$	99.6255	99.6265	99.6022	99.6156
Average	99.6242	99.6140	99.6188	99.6127
Average of all	99.6174	-	-	-

Table 4: Histogram variance findings for cipher images

Algorithm	Lena	Baboon	House	Starfish	Average
Proposed	268.4141	263.6250	275.9922	255.9844	266.0039
Ref. [47]	273.300				

Table 5: Results of metric correlation coefficient

Image	Component	Correlation direction		
		Horizontal	Vertical	Diagonal
Plain image of Lena	Red	0.9551	0.9263	0.9154
	Green	0.9503	0.9365	0.9276
	Blue	0.9423	0.9235	0.8721
Encrypted Lena image	Red	0.0035	0.0081	-0.0042
	Green	-0.0052	0.0072	0.0045
	Blue	0.0052	-0.0073	0.0034

Table 6: Comparative assessment of correlation coefficients across different encryption methods

Image	Encryption scheme	Correlation direction		
		Horizontal	Vertical	Diagonal
Plaintext image of Lena		0.9492	0.9288	0.9050
Encrypted Lena image	Suggested	0.0012	0.0027	0.0012
	Ref. [17]	0.0013	-0.0009	-0.0023
	Ref. [45]	-6.11×10^{-4}	-5.12×10^{-4}	3.46×10^{-4}
	Ref. [46]	0.001	0.0029	0.0019
	Ref. [7]	0.0007	0.0031	-0.0051
	Ref. [35]	0.0156	0.0156	0.0027
	Ref. [5]	0.0095	0.0004	0.0006
	Ref. [18]	-0.0013	0.0006	0.0018

Table 7: The outcomes of the analysis on information entropy

Encryption algorithms	Images	Size	Original	Encrypted
Suggested	Lena	256 × 256	7.2507	7.9975
	Baboon	256 × 256	7.6942	7.9976
	House	256 × 256	6.4311	7.9975
	Starfish	256 × 256	7.8740	7.9975
	Average		7.3125	7.9975
Ref. [45] Ref. [46] Ref. [22] Ref. [7] Ref. [35] Ref. [5] Ref. [18]	Lena	512 × 512	7.6287	7.99929
	Lena	256 × 256		7.9974
	Baboon	512 × 512		7.9993
	Lena	256 × 256		7.9924
	Lena	256 × 256		7.9966
	Lena	512 × 512		7.99941
	Lena	512 × 512		7.999
	Lena	512 × 512		7.9992

further analysis of this relationship, they can deduce the encryption key. The mathematical expressions for these two parameters are as follows:

$$NPCR = \frac{\sum_{s,d} Y(s,d)}{A \times C} \times 100\% \quad (10)$$

The dimensions of the provided image are denoted by A for width and C for height. $Y(s,d)$ can be defined by:

$$Y(s,d) = \begin{cases} 1, & \text{if } Y(s,d) \neq Y'(s,d) \\ 0, & \text{if } Y(s,d) = Y'(s,d). \end{cases} \quad (11)$$

$$UACI = \frac{1}{A \times C} \left[\sum_{s,d} \frac{|U(s,d) - U'(s,d)|}{255} \right] \times 100\% \quad (12)$$

The images denoted as U and U' correspond to encoded/cipher images before and after a single pixel alteration in the plaintext image, respectively. Apart from that, Table 8 displays $UACI$ and $NPCR$ values for the chosen set of four RGB/color images. Across these four color images, mean $NPCR$ value is found to be 99.6097%, while the average $UACI$ value is 33.4982%. These values provide clear evidence of the robustness of the suggested encryption technique against avalanche attacks involving $NPCR$ and $UACI$.

Furthermore, Table 9 offers a comparison of our $NPCR$ and $UACI$ values for the Lena RGB image with those of several other image encryption schemes. It is evident that our proposed encryption scheme outperforms [46, 45, 18, 22] in terms of $NPCR$ and also surpasses [22] concerning $UACI$.

5.6 Analysis of chosen-plaintext, known-plaintext, and ciphertext-only attacks

In cryptanalysis, ciphertext-only, known-plaintext, and chosen-plaintext attacks are commonly considered when

Table 8: Mean npcr and uaci values across diverse images

Images	NPCR(%)	UACI(%)
Lena	99.6277	33.4585
Baboon	99.6094	33.4291
House	99.6028	33.5780
Starfish	99.5987	33.5270
Average	99.6097	33.4982

evaluating the security of a cryptosystem [23]. In a ciphertext-only attack, the adversary has access to one or more ciphertexts and attempts to recover the corresponding plaintexts or obtain information about the secret key. In a known-plaintext attack, the adversary possesses one or more plaintext–ciphertext pairs and attempts to exploit their relationship. A chosen-plaintext attack represents a stronger attack model in which the adversary is assumed to have temporary access to the encryption mechanism and can obtain ciphertexts corresponding to selected plaintexts. Consequently, resistance to chosen-plaintext analysis provides stronger evidence than evaluation under ciphertext-only and known-plaintext scenarios.

To experimentally investigate the resistance of the proposed scheme to chosen-plaintext analysis, special images with highly predictable pixel distributions are considered. First, an all-black image, shown in Figure 9a, is selected as the plaintext and encrypted using the proposed scheme to obtain the corresponding cipher image shown in Figure 9b. The adversary is assumed to exploit the known structure of this plaintext–ciphertext pair in an attempt to derive information equivalent to the chaotic data or keystream employed during encryption. The recovered information is subsequently used in an attempt to decrypt the ciphertext corresponding to another arbitrary plaintext image, namely Lena, shown in Figure 9c. As illustrated in Figure 9d, this attempt does not successfully recover the Lena image, indi-

Table 9: Results of nPCR and uaci, their mean, and a comparison on the lena image

Algorithm	Image	Average NPCR(%)	Average UACI(%)
Our Algorithm	Lena	99.6277	33.4585
Ref. [46]	Lena	99.6040	33.4736
Ref. [45]	Lena	99.5999	33.5060
Ref. [18]	Lena	99.6213	33.4468
Ref. [22]	Lena	99.61	33.78

cating that information obtained from the specially selected black image cannot be directly reused to decrypt a different ciphertext.

The experiment is repeated using an all-white image as another chosen-plaintext input, as illustrated in Figures 9e–9h. Again, the information obtained from the selected plaintext–ciphertext pair does not enable successful recovery of the independently encrypted target image. These experimental results indicate that the encryption output is sufficiently dependent on the input and associated encryption parameters to prevent direct reuse of information extracted from the tested special plaintexts. Therefore, under the considered experimental attack model, the proposed scheme demonstrates resistance to the evaluated chosen-plaintext scenario and, consequently, provides evidence of robustness against the weaker known-plaintext and ciphertext-only scenarios.

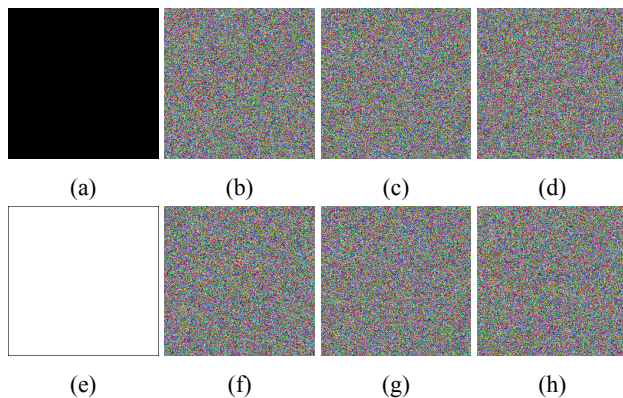


Figure 9: Using black and white images chosen plaintext attack analysis: (a) black image; (b) cipher black image; (c) cipher lena image; (d) decrypted lena image with possible secret key from black image; (e) white image; (f) cipher white image; (g) cipher lena image; (h) decrypted lena image with possible secret key from white image.

5.7 PSNR analysis

PSNR being a yet another metric, and standing for Peak Signal-to-Noise Ratio, is also employed by the image cryptographers to measure the optimal distinction between two images: the input plain image and resulting encrypted image. Its mandate is to assess dissimilarity between these

images. The mathematical formulation associated with this concept is defined in [50].

$$\begin{cases} PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) dB \\ MSE = \frac{1}{M \times N} \sum_{j=1}^N \sum_{l=1}^U (P_0(j, l) - P_1(j, l))^2 \end{cases} \quad (13)$$

The variables used in the equation above are explained as follows:

- (N, U) : The dimensions of the image.
- $P_0(j, l)$: Pixel value of the original image at position (j, l) .
- $P_1(j, l)$: Pixel value of the encrypted image at position (j, l) .
- MSE : Mean Squared Error.

Larger values of the security parameter MSE and lesser values of PSNR indicate more effective encryption and hence improved security. Table 10 presents the PSNR values obtained by different studies. In this table, ‘O-C’ refers to PSNR metric value between cipher and plain images, while ‘O-D’ represents the PSNR between the original and decrypted images. The table reveals that PSNR values are consistently infinite (∞) for the original and decrypted images, implying that MSE equals 0. This observation leads to the conclusion that suggested encryption scheme is lossless. Furthermore, suggested algorithm achieves superior results in the PSNR metric for the Lena image when it is compared to the different works [50, 34, 42, 48].

5.8 Mean absolute error (MAE)

This security metric has been designed to assess the disparity between the ciphertext and plaintext images. In the context of enhanced security concerns, a higher value for this metric is preferred. In the language of Mathematics, this is usually expressed as:

$$MAE = \frac{1}{f \times p} \sum_{u=1}^f \sum_{c=1}^p |C(u, c) - P(u, c)| \quad (14)$$

In the above equation for calculating the value of the metric MAE, the variable P denotes plaintext image, and variable C denotes ciphertext image. Additionally, f and p refers to the dimensions (length and width) of the image. Moreover, Table 11 provides the outcomes of this significant metric. Furthermore, this table offers a comparison of our results with those of other schemes mentioned in [48, 44, 34, 31]. Moreover, our scheme has the superior results than the works [48, 44] against the Lena image.

Table 10: The PSNR results

		Lena	Baboon	House	Starfish
Proposed	(O-D)	∞	∞	∞	∞
	(O-C)	7.8074	8.7850	9.8038	8.2633
Ref. [50]	PSNR (O-D)	96.2956			
	PSNR (O-C)	9.0348			
Ref. [34]	PSNR (O-C)	8.6878			
Ref. [42]	PSNR (O-C)	9.0486			
Ref. [48]	PSNR (O-C)	8.1300			

Table 11: The results of MAE

Name of image	MAE
Lena	77.6436
Baboon	76.2506
House	75.2209
Starfish	84.7547
Average	78.4675
Ref. [48]	77.5379
Ref. [44]	74.93
Ref. [34]	82.8419
Ref. [31]	79.2186

5.9 Encryption time

The proposed image cipher was implemented on a system equipped with a Processor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 2419 Mhz, 4 Core(s), 8 Logical Processor(s) Installed Physical Memory (RAM) 8.00 GB System Type x64-based PC. MATLAB R2024a was used as the programming language. Security is the primary concern in any cryptographic endeavor, and while encryption time is of lesser importance compared to security, it remains a crucial factor for cryptographers. Ciphers with shorter response times have more practical applications in the industry. Typically, security and time are inversely related, requiring cryptographers to strike a balance between these two competing demands when designing a cipher.

Table 12 presents the computational time required by the proposed algorithm to encrypt the four test images. An average encryption time of 0.4269 seconds was obtained under the specified experimental environment, indicating the computational feasibility of the proposed scheme. The reported execution time is also compared with that presented in [46, 17, 22, 21, 11, 28, 32]. The proposed method shows a lower execution time against five studies in this comparison, the values reported across different studies should be interpreted with caution, since they may have been obtained using different processors, software platforms, programming environments, image dimensions, and implementation strategies. Therefore, the comparison with these studies is intended to provide an indicative performance refer-

ence rather than a strict controlled benchmark.

Another important metric for evaluating the computational performance of an encryption scheme is encryption throughput (ET). Encryption throughput represents the amount of image data that can be encrypted per unit time and therefore provides an indication of the processing efficiency of the cipher. A higher throughput value indicates that a larger amount of data can be encrypted within a given time interval. It is mathematically expressed as

$$ET = \frac{Image_{size} \text{ (bits)}}{Encryption_{time} \text{ (seconds)}} \quad (15)$$

The encryption throughput values obtained for the proposed scheme are reported in the fifth column of Table 12. The experimental results show that the proposed cipher achieves favorable throughput. These results provide further evidence of the computational efficiency of the proposed encryption scheme.

5.10 Computational and memory complexity analysis

In addition to the experimental execution time analysis, the computational and memory requirements of the major stages of the proposed cipher are analyzed theoretically. Let the input RGB image have dimensions $m \times n \times 3$ and let

$$N = 3mn \quad (16)$$

denote the total number of color-component values. Furthermore, let

$$d = \min(m, n) \quad (17)$$

denote the maximum possible length, in asymptotic terms, of a diagonal traversed during a diagonal-strip swap-operation.

The proposed encryption procedure consists mainly of plaintext-dependent initialization, chaotic sequence generation, construction of control sequences, diagonal-strip permutation, mask generation, and XOR-based diffusion. Table 13 summarizes the time and memory complexities associated with these stages.

Table 12: Algorithm's computational/encryption speed, throughput and comparison with other schemes

Algorithm	Image name	Image size	Speed (sec)	Mbit/sec
Suggested	Lena	256×256	0.4254	3.8322
	Baboon	256×256	0.4034	4.2442
	House	256×256	0.4267	3.2253
	Starfish	256×256	0.4521	3.7327
	Average		0.4269	3.7586
Ref. [46]	Baboon		0.9447	
Ref. [17]	-		0.54	
Ref. [22]	Lena		0.38	
Ref. [21]	Lena		2.5607	
Ref. [11]	Lena		3.1143	
Ref. [28]	-		1.48	
Ref. [32]	-		0.067230	

Finite-precision arithmetic may affect the behavior of digitally implemented chaotic systems because rounding and quantization can cause deviations from their ideal continuous-precision trajectories and may eventually introduce periodic behavior. This issue can become more pronounced on resource-constrained embedded processors employing reduced numerical precision. The present implementation uses MATLAB double-precision arithmetic; therefore, a detailed investigation of fixed- and reduced-precision implementations of the proposed ILM-based cipher is beyond the scope of this study and will be considered during future embedded-platform implementation.

6 Discussion

The experimental results indicate that the proposed encryption scheme provides a reasonable balance between security and computational simplicity. The four RGB test images, namely Lena, Baboon, House, and Starfish, possess different textures and statistical characteristics, yet the obtained security measures remain relatively consistent. The average information entropy is approximately 7.9975, while the mean NPCR and UACI values are 99.6097% and 33.4982%, respectively. Moreover, the correlation coefficients of the cipher images remain close to zero. These results suggest that the proposed method is not strongly dependent on the particular statistical structure of a single test image.

The main methodological novelty of the proposed scheme lies in its cross-channel diagonal-strip permutation mechanism. Unlike conventional row-, column-, or position-based permutation approaches, the proposed Diagonal Strips Based Scrambler (DSBS) dynamically selects two RGB components and exchanges diagonally oriented pixel strips according to chaotic control sequences. Therefore, the confusion process affects both the spatial arrangement of pixels and their distribution across color channels. The Intertwining Logistic Map further controls the selec-

tion of channels, directions, positions, and diffusion mask, while the plaintext-dependent initialization causes different images to generate different chaotic control sequences.

This additional structural complexity introduces a computational trade-off. Chaotic sequence generation, control-sequence construction, mask formation, and XOR diffusion require $O(mn)$ time for an $m \times n$ RGB image. The DSBS stage is computationally dominant because it performs $3mn$ scrambling iterations, each of which may traverse a diagonal of length proportional to $\min(m, n)$. Hence, its worst-case complexity is

$$O(mn \min(m, n)), \quad (18)$$

which becomes $O(n^3)$ for a square $n \times n$ image. Nevertheless, the operations involved are primarily simple pixel swaps and XOR operations rather than computationally expensive transforms or optimization procedures. The average encryption time of 0.4269 seconds obtained for the 256×256 RGB images indicates that the proposed design remains computationally manageable in the current software implementation.

The memory requirement of the proposed cipher grows linearly with image size because the dominant data structures, including the RGB image planes, chaotic sequences, control arrays, mask, scrambled image, and cipher image, all contain a number of elements proportional to mn . Consequently, the overall memory complexity remains $O(mn)$. This linear growth is favorable for resource-conscious applications, although actual RAM and energy consumption on embedded hardware still require direct experimental validation.

The practical interpretation of the results should also consider the experimental platform. The current implementation was evaluated using MATLAB R2024a on an 11th-generation Intel Core i5 system with 8 GB RAM, rather than on a resource-constrained IoT processor. Therefore, the reported execution time and throughput demonstrate the computational characteristics of the proposed algorithm under a

Table 13: Computational and memory complexity of the major stages of the proposed cipher

Stage	Principal Operations	Time Complexity	Memory Complexity
Plaintext-dependent initialization	Summation of the $3mn$ RGB component values to modify the initial chaotic parameter	$O(mn)$	$O(1)$ auxiliary
Chaotic sequence generation	Generation of the ILM sequences x , y , and z for $3mn$ iterations	$O(mn)$	$O(mn)$
Control-sequence and mask generation	Generation of <i>choice</i> , <i>direction</i> , p , q , and <i>mask</i> from the chaotic sequences over $3mn$ iterations	$O(mn)$	$O(mn)$
RGB channel decomposition	Extraction of the red, green, and blue image planes	$O(mn)$	$O(mn)$
Diagonal-strip permutation (DSBS)	$3mn$ calls to the diagonal-strip swapping routine; each call processes at most $O(d)$ pixels, where $d = \min(m, n)$	$O(mnd)$	$O(mn)$
Mask formation	Organization of the generated mask sequence into an $m \times n \times 3$ mask image	$O(mn)$	$O(mn)$
XOR diffusion	One XOR operation for each of the $3mn$ RGB component values	$O(mn)$	$O(mn)$
Overall encryption	Chaotic generation + control generation + DSBS + diffusion	$O(mnd)$	$O(mn)$
Overall decryption	Mask regeneration + inverse XOR + reverse-order diagonal swapping	$O(mnd)$	$O(mn)$

conventional software environment, but they do not directly establish real-time performance or energy efficiency on embedded hardware. Evaluation on representative platforms such as Raspberry Pi or ARM-based processors would provide stronger evidence regarding practical IoT deployment.

The proposed cipher is based on symmetric-key cryptography, where the initial conditions and control parameters of the chaotic system act as secret-key components. In a practical deployment, these parameters should be securely provisioned or established through an appropriate key-management mechanism rather than transmitted over an unsecured channel. Reuse of identical key parameters over a large number of sessions should also be avoided. Although the plaintext-dependent initialization helps modify the generated chaotic sequences for different images, periodic key renewal would remain advisable in a real-world implementation.

The results reported in this study are obtained mainly from standard RGB benchmark images. While their different textures provide some evidence of consistency, broader validation using application-specific images, including surveillance, medical, industrial, and other IoT-generated visual data, would strengthen the generalization claims. Similarly, the influence of image resolution and color depth deserve further study. The current results indicate that the statistical randomization capability is maintained for the tested image sizes, but larger-scale experiments would provide a more complete assessment of scalability.

Overall, the proposed combination of ILM-controlled diagonal-strip permutation and XOR diffusion offers a distinctive image-encryption framework with competitive statistical and differential security characteristics. Its principal advantage lies in introducing dynamic cross-channel spatial transformations while retaining relatively simple primitive operations. Future work will therefore focus on embedded implementation, reduced-precision arithmetic, broader image datasets, and systematic evaluation across multiple image resolutions and color depths.

7 Conclusion

This paper presented a lightweight chaos-based RGB image encryption scheme designed for resource-conscious image security applications, with particular relevance to IoT environments. The proposed framework combines Inter-twining Logistic Map (ILM)-based chaotic control with a dynamic cross-channel diagonal-strip permutation mechanism and XOR-based diffusion. Unlike conventional row- or column-oriented scrambling strategies, the proposed confusion stage repeatedly exchanges chaos-controlled diagonal strips between dynamically selected RGB components, thereby disturbing both the spatial arrangement and cross-channel distribution of image pixels.

The experimental results demonstrate that the proposed scheme achieves strong statistical and differential characteristics. The average information entropy approaches the

ideal value of 8, while the obtained NPCR and UACI values are close to their expected theoretical levels. The correlation coefficients of the cipher images remain close to zero, and the key-sensitivity analysis confirms that very small perturbations in the secret-key parameters produce substantially different encryption outputs. In addition, the reconstruction analysis verifies lossless decryption, with zero MSE and infinite PSNR between the original and recovered images. The proposed scheme also achieves an average encryption time of 0.4269 seconds for the tested 256×256 RGB images under the specified conventional computing environment.

The theoretical analysis shows that chaotic sequence generation and XOR diffusion require $O(mn)$ time, whereas the diagonal-strip permutation dominates the computational cost with worst-case complexity $O(mn \min(m, n))$. The overall memory requirement remains $O(mn)$, indicating linear memory growth with image dimensions. Although these results support the computational feasibility and security potential of the proposed design, the current implementation has been evaluated on a conventional computing platform rather than directly on embedded IoT hardware. Therefore, practical deployment claims should be interpreted in this context.

Future work will focus on implementing the proposed cipher on resource-constrained platforms such as Raspberry Pi and ARM-based processors, where execution time, memory utilization, energy consumption, and finite-precision effects can be investigated directly. Additional evaluation using larger and more diverse image collections, multiple image resolutions and color depths, as well as application-specific images such as surveillance, medical, and industrial imagery, will further establish the scalability and generalization capability of the proposed framework.

Source code availability

The source code used to implement the proposed encryption and decryption scheme is available from the corresponding author upon reasonable request.

References

- [1] Oluwakemi Christiana Abikoye, Ahmad Dokoro Haruna, Abdullahi Abubakar, Noah Oluwatobi Akande, and Emmanuel Oluwatobi Asani. Modified advanced encryption standard algorithm for information security. *Symmetry*, 11(12):1484, 2019. <https://doi.org/10.3390/sym11121484>.
- [2] Mohammed AbuTaha, Mousa Farajallah, Radwan Tahboub, and Mohammad Odeh. Survey paper: cryptography is the science of information security. 2011.
- [3] Musheer Ahmad, Shafali Agarwal, Ahmed Alkhayyat, Adi Alhudhaif, Fayadh Alenezi, Amjad Hussain Zahid, and Nojood O Aljehane. An image encryption algorithm based on new generalized fusion fractal structure. *Information Sciences*, 592:1–20, 2022. <https://doi.org/10.1016/j.ins.2022.01.042>.
- [4] Erhan Akbal, Prabal Datta Barua, Sengul Dogan, Turker Tuncer, and U Rajendra Acharya. Despat-net25: Data encryption standard cipher model for accurate automated construction site monitoring with sound signals. *Expert Systems with Applications*, 193:116447, 2022. <https://doi.org/10.1016/j.eswa.2021.116447>.
- [5] Wassim Alexan, Dina El-Damak, and Mohamed Gabr. Image encryption based on fourier-dna coding for hyperchaotic chen system, chen-based binary quantization s-box, and variable-base modulo operation. *IEEE Access*, 12:21092–21113, 2024. <https://doi.org/10.1109/ACCESS.2024.3363018>.
- [6] Wassim Alexan, Marwa Elkandoz, Maggie Mashaly, Eman Azab, and Amr Aboshousha. Color image encryption through chaos and kaa map. *Ieee Access*, 11:11541–11554, 2023. <https://doi.org/10.1109/ACCESS.2023.3242311>.
- [7] Wassim Alexan, Marwa Elkandoz, Maggie Mashaly, Eman Azab, and Amr Aboshousha. Color image encryption through chaos and kaa map. *Ieee Access*, 11:11541–11554, 2023. <https://doi.org/10.1109/ACCESS.2023.3242311>.
- [8] Wassim Alexan, Mohamed Youssef, Hisham H Hussein, Karim K Ahmed, Khalid M Hosny, Abdallah Fathy, and Marvy Badr Monir Mansour. A new multiple image encryption algorithm using hyperchaotic systems, svd, and modified rc5. *Scientific Reports*, 15(1):9775, 2025. <https://doi.org/10.1038/s41598-025-92065-x>.
- [9] Feng-Ping An and Jun-e Liu. Image encryption algorithm based on adaptive wavelet chaos. *Journal of Sensors*, 2019:1–12, 2019. <https://doi.org/10.1155/2019/2768121>.
- [10] Pulung Nurtantio Andono et al. Improved pixel and bit confusion-diffusion based on mixed chaos and hash operation for image encryption. *IEEE Access*, 10:115143–115156, 2022. <https://doi.org/10.1109/ACCESS.2022.3218886>.
- [11] Zia Bashir, Nadeem Iqbal, and Muhammad Hanif. A novel gray scale image encryption scheme based on pixels' swapping operations. *Multimedia Tools and Applications*, 80:1029–1054, 2021. <https://doi.org/10.1007/s11042-020-09695-8>.
- [12] Anupkumar M Bongale, Kishore Bhamidipati, Arunkumar M Bongale, and Satish Kumar. Hybrid

- international data encryption algorithm for digital image encryption. In *Proceedings of International Conference on Sustainable Expert Systems: ICSES 2020*, pages 465–477. Springer, 2021. https://doi.org/10.1007/978-981-33-4355-9_35.
- [13] Sumit Singh Dhanda, Brahmjit Singh, and Poonam Jindal. Lightweight cryptography: a solution to secure iot. *Wireless Personal Communications*, 112:1947–1980, 2020. <https://doi.org/10.1007/s11277-020-07134-3>.
- [14] Zhou-quan Du, Qi-jian Xu, Jie Zhang, and Mu Li. Design and analysis of dynamic s-box based on feistel. In *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 590–594. IEEE, 2015. <https://doi.org/10.1109/IAEAC.2015.7428622>.
- [15] Marwa Tarek Elkandoz and Wassim Alexan. Image encryption based on a combination of multiple chaotic maps. *Multimedia Tools and Applications*, 81(18):25497–25518, 2022. <https://doi.org/10.1007/s11042-022-12595-8>.
- [16] Runchen Gao, Shen Li, Yuqi Gao, and Rui Guo. A lightweight cryptographic algorithm for the transmission of images from road environments in self-driving. *Cybersecurity*, 4:1–11, 2021. <https://doi.org/10.1186/s42400-020-00066-2>.
- [17] Xinyu Gao, Jun Mou, Santo Banerjee, Yinghong Cao, Li Xiong, and Xiaoyang Chen. An effective multiple-image encryption algorithm based on 3d cube and hyperchaotic map. *Journal of King Saud University-Computer and Information Sciences*, 34(4):1535–1551, 2022. <https://doi.org/10.1016/j.jksuci.2022.01.017>.
- [18] Seong Oun Hwang, Hafiz Muhammad Waseem, and Noor Munir. Billiard quantum chaos: A pioneering image encryption scheme in the post-quantum era. *IEEE Access*, 12:85150–85164, 2024. <https://doi.org/10.1109/ACCESS.2024.3415083>.
- [19] Nadeem Iqbal. Image encryption using queen. *Multimedia Tools and Applications*, pages 1–35, 2023. <https://doi.org/10.1007/s11042-023-15674-6>.
- [20] Nadeem Iqbal, Sagheer Abbas, Muhammad Adnan Khan, Tahir Alyas, Areej Fatima, and Aiesha Ahmad. An rgb image cipher using chaotic systems, 15-puzzle problem and dna computing. *IEEE Access*, 7:174051–174071, 2019. <https://doi.org/10.1109/ACCESS.2019.2956389>.
- [21] Nadeem Iqbal, Muhammad Hanif, Zia Ul Rehman, and Muhammad Zohaib. On the novel image encryption based on chaotic system and dna computing. *Multimedia Tools and Applications*, 81(6):8107–8137, 2022. <https://doi.org/10.1007/s11042-022-11912-5>.
- [22] Bhat Jasra and Ayaz Hassan Moon. Color image encryption and authentication using dynamic dna encoding and hyper chaotic system. *Expert Systems with Applications*, 206:117861, 2022. <https://doi.org/10.1016/j.eswa.2022.117861>.
- [23] Shuming Jiao, Ting Lei, Yang Gao, Zhenwei Xie, and Xiaocong Yuan. Known-plaintext attack and ciphertext-only attack for encrypted single-pixel imaging. *IEEE Access*, 7:119557–119565, 2019. <https://doi.org/10.1109/ACCESS.2019.2936119>.
- [24] Sanjay Kumar and Deepmala Sharma. A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm: S. kumar, d. sharma. *Artificial intelligence review*, 57(4):87, 2024. <https://doi.org/10.1007/s10462-024-10719-0>.
- [25] Qiang Lai, Genwen Hu, Uğur Erkan, and Abdurrahim Toktas. A novel pixel-split image encryption scheme based on 2d salomon map. *Expert Systems with Applications*, 213:118845, 2023. <https://doi.org/10.1016/j.eswa.2022.118845>.
- [26] Navdeep Lata and Raman Kumar. Dsit: A dynamic lightweight cryptography algorithm for securing image in iot communication. *International Journal of Image and Graphics*, 23(04):2350035, 2023. <https://doi.org/10.1142/S0219467823500353>.
- [27] Lizong Li. A self-reversible image encryption algorithm utilizing a novel chaotic map. *Nonlinear Dynamics*, 113(7):7351–7383, 2025. <https://doi.org/10.1007/s11071-024-10726-4>.
- [28] Xiaofeng Liao, Ayesha Kulsoom, and Sami Ullah. A modified (dual) fusion technique for image encryption using sha-256 hash and multiple chaotic maps. *Multimedia Tools and Applications*, 75(18):11241–11266, 2016. <https://doi.org/10.1007/s11042-015-2851-7>.
- [29] Fawad Masood, Wadii Boulila, Abdullah Alsaedi, Jan Sher Khan, Jawad Ahmad, Muazzam A Khan, and Sadaqat Ur Rehman. A novel image encryption scheme based on arnold cat map, newton-leipnik system and logistic gaussian map. *Multimedia Tools and Applications*, 81(21):30931–30959, 2022. <https://doi.org/10.1007/s11042-022-12844-w>.
- [30] Mahmoud Mohamed, Muhammad Moinuddin, and Abdulllah Balamash. Chaoticmine: hybrid asymmetric autoencoder and chaotic map framework for secure

- wireless image encryption and threat detection. *Journal of King Saud University Computer and Information Sciences*, 38(7):686, 2026. <https://doi.org/10.1007/s44443-026-01159-7>.
- [31] Hira Nazir, Imran Sarwar Bajwa, Saima Abdullah, Rafaqut Kazmi, and Muhammad Samiullah. A color image encryption scheme combining hyperchaos and genetic codes. *IEEE Access*, 10:14480–14495, 2022. <https://doi.org/10.1109/ACCESS.2022.3143096>.
- [32] Tsafack Nestor, Nkapkop Jean De Dieu, Kengne Jacques, Effa Joseph Yves, Abdullah M Iliyasu, and Ahmed A Abd El-Latif. A multidimensional hyperjerk oscillator: Dynamics analysis, analogue and embedded systems implementation, and its application as a cryptosystem. *Sensors*, 20(1):83, 2019. <https://doi.org/10.3390/s20010083>.
- [33] Ying Niu, Zheng Zhou, and Xuncaizhang. An image encryption approach based on chaotic maps and genetic operations. *Multimedia Tools and Applications*, 79:25613–25633, 2020. <https://doi.org/10.1007/s11042-020-09237-2>.
- [34] Benyamin Norouzi and Sattar Mirzakuchaki. A fast color image encryption algorithm based on hyper-chaotic systems. *Nonlinear Dynamics*, 78(2):995–1015, 2014. <https://doi.org/10.1007/s11071-014-1492-0>.
- [35] Esau Taiwo Oladipupo, Oluwakemi Christiana Abikoye, and Joseph Bamidele Awotunde. A lightweight image cryptosystem for cloud-assisted internet of things. *Applied Sciences*, 14(7):2808, 2024. <https://doi.org/10.3390/app14072808>.
- [36] Vinod Patidar, NK Pareek, G Purohit, and KK Sud. A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. *Optics communications*, 284(19):4331–4339, 2011. <https://doi.org/10.1016/j.optcom.2011.05.028>.
- [37] K Abhimanyu Kumar Patro and Bibhudendra Acharya. An efficient colour image encryption scheme based on 1-d chaotic maps. *Journal of Information Security and Applications*, 46:23–41, 2019. <https://doi.org/10.1016/j.jisa.2019.02.006>.
- [38] Padmapriya Praveenkumar, Rengarajan Amirtharajan, Karuppuswamy Thenmozhi, and John Bosco Balaguru Rayappan. Fusion of confusion and diffusion: a novel image encryption approach. *Telecommunication Systems*, 65:65–78, 2017. <https://doi.org/10.1007/s11235-016-0212-0>.
- [39] Muhammad Rana, Quazi Mamun, and Rafiqul Islam. Lightweight cryptography in iot networks: A survey. *Future Generation Computer Systems*, 129:77–89, 2022. <https://doi.org/10.1016/j.future.2021.11.011>.
- [40] Usman Shahid, Shamsa Kanwal, Mahwish Bano, Saba Inam, Manal Elzain Mohamed Abdalla, and Zafar Ahmed Shaikh. Blockchain driven medical image encryption employing chaotic tent map in cloud computing. *Scientific Reports*, 15(1):6236, 2025. <https://doi.org/10.1038/s41598-025-90502-5>.
- [41] Petra Snaselova and Frantisek Zboril. Genetic algorithm using theory of chaos. *Procedia computer science*, 51:316–325, 2015. <https://doi.org/10.1016/j.procs.2015.05.248>.
- [42] Nidhi Taneja, Balasubramanian Raman, and Indra Gupta. Combinational domain encryption for still visual data. *Multimedia Tools and Applications*, 59:775–793, 2012. <https://doi.org/10.1007/s11042-011-0775-4>.
- [43] Ye Tian, Zhimao Lu, et al. Chaotic s-box: Intertwining logistic map and bacterial foraging optimization. *Mathematical Problems in Engineering*, 2017, 2017. <https://doi.org/10.1155/2017/6969312>.
- [44] Asad Ullah, Awais Aziz Shah, Jan Sher Khan, Mazhar Sajjad, Wadii Boulila, Akif Akgul, Junaid Masood, Fuad A Ghaleb, Syed Aziz Shah, Jawad Ahmad, et al. An efficient lightweight image encryption scheme using multichaos. *Security and Communication Networks*, 2022, 2022. <https://doi.org/10.1155/2022/5680357>.
- [45] Talha Umar, Mohammad Nadeem, and Faisal Anwer. Chaos based image encryption scheme to secure sensitive multimedia content in cloud storage. *Expert Systems with Applications*, 257:125050, 2024. <https://doi.org/10.1016/j.eswa.2024.125050>.
- [46] Mingxu Wang, Lin Teng, Wenjie Zhou, Xiaopeng Yan, Zhiqiu Xia, and Shuang Zhou. A new 2d cross hyperchaotic sine-modulation-logistic map and its application in bit-level image encryption. *Expert Systems with Applications*, 261:125328, 2025. <https://doi.org/10.1016/j.eswa.2024.125328>.
- [47] Xingyuan Wang, Yining Su, Chao Luo, Fuzhong Nian, and Lin Teng. Color image encryption algorithm based on hyperchaotic system and improved quantum revolving gate. *Multimedia Tools and Applications*, 81(10):13845–13865, 2022. <https://doi.org/10.1007/s11042-022-12220-8>.
- [48] Xiangjun Wu, Kunshu Wang, Xingyuan Wang, Haibin Kan, and Jürgen Kurths. Color image dna

encryption using nca map-based cml and one-time keys. *Signal Processing*, 148:272–287, 2018. <https://doi.org/10.1016/j.sigpro.2018.02.028>.

- [49] Yue Wu, Joseph P Noonan, Sos Agaian, et al. Npcr and uaci randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, 1(2):31–38, 2011.
- [50] Congxu Zhu. A novel image encryption scheme based on improved hyperchaotic sequences. *Optics communications*, 285(1):29–37, 2012. <https://doi.org/10.1016/j.optcom.2011.08.079>.