Engineering Reports

WILEY

Engineering Reports Open Access

RESEARCH ARTICLE OPEN ACCESS

A Randomized Non-overlapping Encryption Scheme for Enhanced Image Security in Internet of Things (IoT) Applications

Muhammad Aqeel[1] | Arfan Jaffar[1] | Muhammad Faheem[2,3] | Muhammad Waqar Ashraf[4] | Nadeem Iqbal[5] | Shahid Yousaf[5] | Hossam Diab[6,7]

[1]Department of Computer Science & IT, Superior University, Lahore, Pakistan | [2]School of Technology and Innovations, University of Vaasa, Vaasa, Finland | [3]VTT-Technical Research Center of Finland, Espoo, Finland | [4]Department of Computer Engineering, Bahauddin Zakariya University, Multan, Pakistan | [5]Department of Computer Science & IT, The University of Lahore, Lahore, Pakistan | [6]Math and Computer Science Department, Faculty of Science, Menoufia University, Menoufia, Egypt | [7]Computer Science Department, Applied College, Taibah University, Khaybar, Saudi Arabia

Correspondence: Muhammad Faheem (muhammfa@uwasa.fi)

Received: 11 September 2024 | **Revised:** 3 December 2024 | **Accepted:** 7 December 2024

Funding: This research work of M. Faheem is supported by VTT-Technical Research Centre of Finland.

Keywords: chaos | cyber security | decryption | encryption | image processing | IoT | secret key

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has underscored the critical need to safeguard the data they store and transmit. Among various data types, digital images often carry highly sensitive information, making their protection against breaches essential. This study introduces a novel image encryption algorithm specifically designed to bolster the security of images in resource-constrained IoT ecosystems. Leveraging the randomness of a 5D multi-wing hyperchaotic map, the proposed method employs pairs of non-overlapping rectangles to induce confusion by swapping the pixels they encompass. Repeated iterations of this operation achieve significant confusion effects, enhancing encryption strength. To validate the robustness of the proposed algorithm, standard benchmark images were utilized, and rigorous security metrics —including information entropy, correlation coefficient, histogram uniformity, and resistance to differential attacks —were analyzed. Results demonstrate that the algorithm not only ensures strong protection against unauthorized access but also maintains low computational complexity, making it ideal for IoT applications. This research provides a foundational step toward ensuring the confidentiality and integrity of visual data in an increasingly interconnected digital world.

1 | Introduction

The advent of the IoT has revolutionized the way we interact with technology, integrating devices into our daily lives in unprecedented ways. These IoT devices span smart home appliances to industrial sensors and healthcare monitors. Besides, these devices are creating a highly interconnected world where data flows seamlessly between devices and systems. No doubt, digital images play a particularly crucial role among these types of data generated and transmitted by IoT devices. Additionally, these images are being employed in varied real-world applications like remote sensing [1], security surveillance [2], medical imaging [3], and smart city infrastructure [4]. However, the sensitive nature of these images poses significant security and privacy challenges [5].

The deployment of IoT devices in varied civic amenities has led to an increased risk of cyber-attacks [6]. Apart from that, different adversaries and hackers are targeting the confidentiality and data

integrity of the information these devices carry. In particular, these devices are capturing the images perennially, and sometimes these images contain the personally identifiable information (PII) [7]. If these images are accessed in an unauthorized way, severe consequences may happen, and in the worst case, it may threaten national security. For the sake of an example, in different defense and military applications, drones capture the aerial images. These images may contain the strategic information. In the same fashion, in varied healthcare settings, medical imaging data must be protected to ensure patient confidentiality and to comply with regulatory standards [8].

Given the strategic value and sensitivity of images in various IoT applications, it is imperative to develop robust encryption mechanisms that can safeguard this data from unauthorized access and manipulation. Traditional encryption algorithms like DES [9], AES [10], RSA [11], while effective, often fall short in the IoT context due to the limited computational resources and power constraints of IoT devices. This necessitates the development of lightweight, efficient, and highly secure encryption algorithms tailored specifically for IoT environments.

An objective study of the literature of image cryptography reveals that many mathematical constructs have been employed by the security analysts to come up with novel and robust image encryption algorithms like chaotic systems [12–14], cellular automatons [15–18], latin square [19], latin cube [20], magic square [21], Sudoku [22, 23], Pascal's matrix [24], rectangle [25] to name a few.

The work [15] employed a 3-layered architecture dubbed as diffusion-permutation-diffusion. Since the image pixels are highly interrelated with each other, the initial diffusion installed in the reported work causes a redistribution of pixel intensities by assigning a new value for all the pixels of the given image. Besides, permutation inspired by the Arnold cat's map has been carried out, due to which strong correlation among the pixels got dismantled. To heighten the security of the proposed cryptographic work, DNA encoding has also been implemented. Besides, the work [19] developed an adaptive image cryptosystem using the Latin squares and fractional order Lü system. The authors of this work contend that their cipher is immune from the twin attacks of chosen plain image attacks (CPA) and known plain image attacks (KPA). The modus operandi of this work proceeds like this. The sum of bit planes and pixel values of the plaintext image was utilized for the selection of chaotic sequences and the Latin squares. This approach proves very effective to thwart the CPA and KPA. Apart from that, through the usage of the fractional order Lü system, the algorithm spawned self-orthogonal Latin squares for confusing bit planes of the given plaintext image at the bit-level. This process was followed by a diffusion process at the pixel level. In yet another work [21], all order-4 (4×4) magic squares have been employed to boost the security effects in the potential image cipher. It is to be noted that the work exploited all the 880 magic squares in the work [21]. The aforementioned myriads of magic squares were taken in an arbitrary way to complete the scrambling algorithm for the given scheme. Besides, the Intertwining Logistic Map (ILM) was selected to get chaotic data.

Moreover, the study [25] utilized the mathematical construct of a rectangle for designing the scrambling algorithm to come up with a new image cipher. Besides, according to the contention of the authors of this work, they have improved the computational time in the given scheme. Moreover, two chaotic maps have been employed to get the arbitrary and chaotic data to be used for the confusion and diffusion operations. To permute the pixels in the given plaintext image, rectangles with different dimensions and sizes were randomly created in the given image. Now the pixels residing at the boundaries of rectangles were rotated clockwise or anti-clockwise for some given amount. All these specifications of the rectangles were obtained through the random numbers. Lastly, a simple XoR operation was carried out to pour the diffusion effects in the image cipher. The simulation and the security analyses rendered very promising results. In particular, the information entropy and the computational time claimed by the authors were 7.9975 and 0.5156 s, respectively.

In the field of cryptography, no doubt, novel ciphers are developed. In sharp contrast to that, there exists an enterprise of cryptanalysis in which loopholes, lacunas, and other inner defects are spotted and fixed to improve the security. Hackers, adversaries, and other persons with malicious intentions exploit these loopholes to materialize their nefarious designs. Image encryption niche is no exception. Many image ciphers have been broken and cracked, as the study of literature indicates. For example, the works [26] and [27] were cracked by [28] and [29].

The encryption algorithm given in [26] was based exclusively on diffusion and permutation operations. In particular, row- and column-wise swapping of the pixels at the bit level for the given image was carried out. Besides, the diffusion operation was done in both the backward and forward directions. The cryptanalytic work [28] was based on the chosen-plaintext attack. In particular, the secret key of the image cipher was obtained by using 258 plain images. The authors of the reported work recommended that thorough cryptanalysis must be carried out apart from the standard statistical evaluations.

An other research work [27] employed two 1D chaotic maps whose random numbers were used for performing the twin operations of diffusion and confusion. After that, the given plaintext image was encrypted through the backward-diffusion, forward-diffusion and the row-column permutation. According to [29], the work couldn't defy the potential chosen plaintext attack. Moreover, the stream of chaotic data in the developed scheme was independent of the plain text image. In this way, the required permutation and diffusion key streams could be obtained by setting the pixels of the chosen plain images. Lastly, the authors of the work [29] claimed that they improved the security effects of the algorithm [27] without enhancing the complexity of the work.

Inspired by the above works, particularly of [25], this study proposes a novel image encryption algorithm designed to enhance the security of images in IoT devices. Our approach leverages the principles of chaotic systems, which are well-known for their high sensitivity to initial conditions and inherent randomness. By generating random, non-overlapping rectangles over the image and interactively swapping pixels within those regions, the proposed algorithm introduces significant scrambling effects that obfuscate the original image content. This

method not only ensures high levels of security but also maintains computational efficiency, making it suitable for deployment in resource-constrained IoT devices.

The proposed algorithm has been rigorously evaluated using standard benchmark images, with a focus on key security metrics such as entropy, correlation coefficient, and histogram uniformity. Additionally, we assess the algorithm's resistance to various cryptographic attacks, including differential and statistical attacks, to demonstrate its robustness. Our results indicate that the proposed encryption scheme effectively secures image data, providing a reliable solution for protecting the privacy and integrity of visual information in IoT applications.

The remaining article has been set like this: Section 2 briefly explains the theory of chaos and its incarnation, the 5D multi-wing hyperchaotic system. Moreover, the suggested work has been given in the Section 3. Apart from that, the Section 4 selects some test images from the image repository to show the robustness and defiance of the suggested scheme for the security of images. A thorough and sweeping security analysis has been done in the Section 5. Finally, Section 6 closes the paper with necessary concluding remarks.

2 | Idea of Chaos and Its Incarnation 5D Multi-wing Hyperchaotic System

The idea of chaos says that a very light change in the initial conditions of some dynamic system may lead to marked changes in the final outcome [30]. This idea is often dubbed as a 'butterfly effect.' Scientists and mathematicians have introduced a plethora of chaotic maps and systems by complying with the idea of chaos. 5D multi-wing hyperchaotic system is one of the incarnations of this marvelous idea. The properties like mixing, ergodicity, randomness, aperiodicity, and unpredictability characterize these maps [31]. These maps exist in different flavors, like 1D, 2D, higher-dimensional, and hyperchaotic maps. The random data given by these maps are normally used for realizing the effects of scrambling and diffusion for the image ciphers. The peculiar algorithmic logic of the study necessitated us to select the 5D multi-wing hyperchaotic map [32]. The mathematical representation of this map is:

$$\begin{aligned} \dot{p} &= -ap + qr \\ \dot{q} &= -bq + ft \\ \dot{r} &= -cr + gs + pq \\ \dot{s} &= ds - hp \\ \dot{t} &= et - p^2 q \end{aligned} \tag{1}$$

In this map, p, q, r, s, t are the initial values. Moreover, the list a, b, c, d, e, f, g, h is characterized as the parameters of the system. Besides, the system contains both the linear and nonlinear terms. Nonlinear terms present in the system are qr, pq, and p^2q. The research work [33] details various properties of this map, such as periodic orbits. A step value of 0.001 was used to plot the different attractors of this chaotic system. Figure 1 illustrates the chaotic behavior of this map. Lastly, the Lyapunov exponents of this system are $\{L_1, L_2, L_3, L_4, L_5\} = \{9.979, 1.96, 0.005362, -19.13, -27.82\}$, as shown in Figure 2.

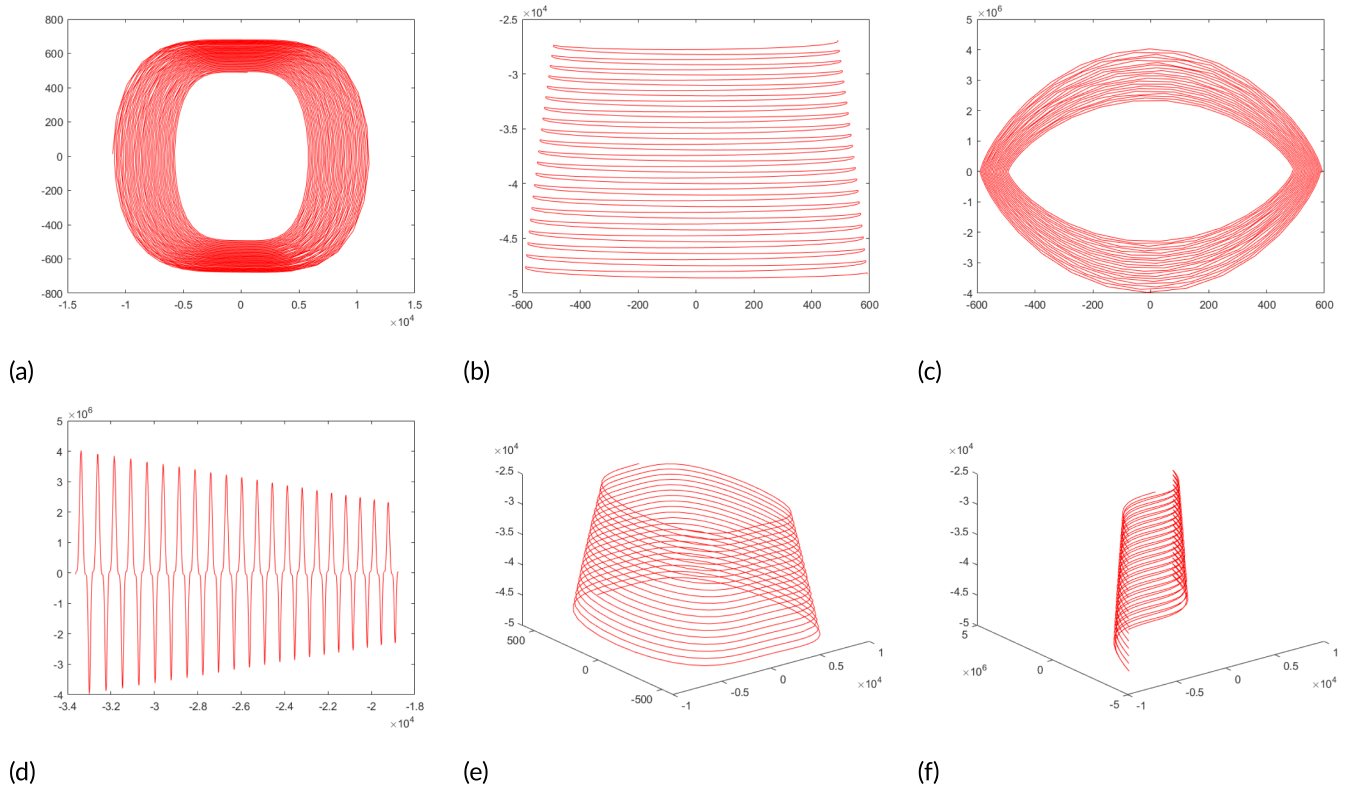


FIGURE 1 | Different attractors of the System (1) in planes and spaces: (a) pq plane; (b) rs plane; (c) qt plane; (d) st plane; (e) 3D view of pqr space; (f) 3D view of ptr space.

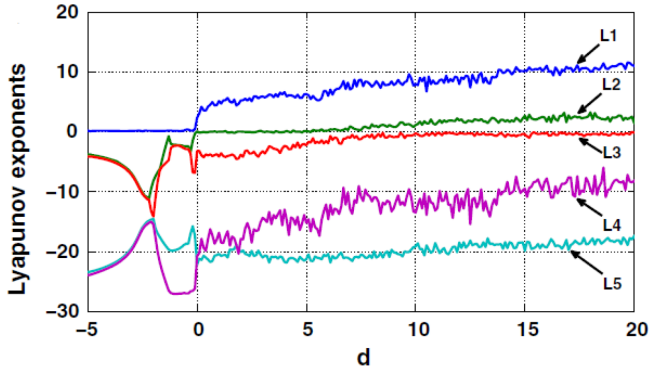


FIGURE 2 | System (1)'s Lyapunov exponents.

ALGORITHM 1 | *Key Streams.*

Input: $Image, p, q, r, s, t, a, b, c, d, e, f, g, h, m, n$
Output: $\{r1_t\}_{t=1}^{3mn}, \{s1_k\}_{k=1}^{3mn}, \{r2_k\}_{k=1}^{3mn}, \{s2_k\}_{k=1}^{3mn}, \{mask_k\}_{k=1}^{3mn}$
1: Spark the Chaotic System (1) with the initial values p, q, r, s, t.
2: $[m, n] \leftarrow size(Image)$
3: **for** $k \leftarrow 1$ to $3mn$ **do**
4: $r_1(k) \leftarrow \lfloor \text{mod}((p(k)) - \lfloor abs(p(k)) \rfloor \times 10^{14}, m) \rfloor + 1$
5: $s_1(k) \leftarrow \lfloor \text{mod}(abs(q(k)) - \lfloor abs(q(k)) \rfloor \times 10^{14}, 3n) \rfloor + 1$
6: $r_2(k) \leftarrow \lfloor \text{mod}(abs(r(k)) - \lfloor abs(r(k)) \rfloor \times 10^{14}, m) \rfloor + 1$
7: $s_2(k) \leftarrow \lfloor \text{mod}(abs(s(k)) - \lfloor abs(s(k)) \rfloor \times 10^{14}, 3n) \rfloor + 1$
8: $mask(k) \leftarrow \lfloor \text{mod}(abs(t(k)) - \lfloor abs(t(k)) \rfloor \times 10^{14}, 256) \rfloor$
9: **end for**

3 | Proposed Scheme for the Image Encryption

This study has conceived a novel idea to scramble the pixels of the given color image. The pairs of randomly generated rectangles have carried out the seminal task of scrambling/confusion. As the color image is input to the system, the sum of all the pixels of the given image is calculated. This sum would serve as the plaintext sensitivity to the potential image cipher. After that, the three planes of red, green, and blue from the color image are extracted. These three planes are merged with each other in a horizontal fashion to come up with a single big gray scale image. Now, a pair of rectangles with the same dimensions but in different locations is spawned in the single gray scale image. Filters have been applied to check whether the rectangles so generated overlap with each other? If they do, nothing is carried out, and the control is transferred to the start of the loop. If they don't, the pixels encompassed by the pair of the rectangles are swapped with each other. This step has been iterated numerous times to pour the scrambling effects into the given gray scale image. The diffusion effects have been realized by the simple XoR operation made between the scrambled image and the mask image of random numbers. Lastly, this big gray scale cipher image is broken down into three components of red, green, and blue. Finally, these three components are merged with each other to form a color cipher image. Four streams of random numbers have been used to determine the top left corners of the pair of rectangles. Interestingly, these four streams have been recycled to come up with the length and the width of the rectangles. Lastly, the final and fifth stream has been utilized to embed the diffusion effects in the required cipher image.

3.1 | Algorithm for the Key Stream Generation

Random and arbitrary data has a very critical value in the realm of cryptography. So, this data must be generated with great care. This section will describe the way the required data has been generated. Let $Image$ be the color image with the dimensions of $m \times n \times 3$. To incorporate plaintext sensitivity into the image cipher, the following equation has been used.

$$p = 1.0 + \frac{sum(sum(sum(Image)))}{2^{40}} \quad (2)$$

where the expression $sum(sum(sum(Image)))$ finds the sum of all the pixel intensity values of three planes of the given image $Image$. Call the Algorithm 1 with the set of values: $Image$, $p = 1.0, q = 1.0, r = 1.0, s = 1.0, t = 1.0, a = 10.0, b = 60.0, c = 20.0, d = 15.0, e = 40.0, f = 1.0, g = 50.0, h = 10.0$. Line 1 sparks the chaotic system (1), which renders the streams p, q, r, s, t of random numbers. These numbers are very primitive and raw and must be customized to make them compatible with the algorithmic logic this study has conceived. Moreover, line 2 finds the size of the color image. The *for* loop at line 3 is iterating for $3mn$ times. Five streams $\{r1_k\}_{k=1}^{3mn}$, $\{s1_k\}_{k=1}^{3mn}$, $\{r2_k\}_{k=1}^{3mn}$, $\{s2_k\}_{k=1}^{3mn}$, $\{mask_k\}_{k=1}^{3mn}$ are being introduced at lines 4, 5, 6, 7, and 8 respectively. Each of these five streams contains the integers in the ranges $[1, 2, \ldots, m]$, $[1, \ldots, 3n]$, $[1, 2, \ldots, m]$, $[1, 2, 3, \ldots, 3n]$, $[0, 1, \ldots, 255]$. The symbol $\lfloor . \rfloor$ represents the floor function. Moreover, the $\text{mod}(g, j)$ operator returns the remainder when integer g is divided by some other integer j.

To ignite the chaotic map, users will provide a predefined encryption key, which serves as the source for generating the required initial values and system parameters. This key can be selected in the form of a password or a numeric value, depending on the implementation. This ensures both ease of use for the user and security by abstracting the complexity of parameter selection.

The process of incorporating plaintext sensitivity has been designed to be fully automated within the encryption algorithm, requiring no manual intervention from the user. When a user inputs a plaintext image for encryption, the algorithm automatically computes the sum of the pixel values and adjusts the initial value of the chaotic system accordingly. This step is seamlessly integrated into the encryption process, ensuring usability for non-technical users. The user is only required to provide the plaintext image and the encryption key, without needing to understand or manage the underlying operations.

3.2 | Proposed Image Encryption Algorithm

Invoke the Algorithm 2 for the encryption of the given color image $Image$. The arguments to this algorithm are $Image$, $r1$, $s1$, $r2$, and $s2$. Here, we explain the working of the proposed algorithm in a step-by-step fashion.

ALGORITHM 2 | Non-Overlapping Rectangles Based Confusion Algorithm.

Input: $Image, r1, s1, r2, s2$
Output: $Combined'$

1: $Red \leftarrow Image(:, :, 1)$
2: $Green \leftarrow Image(:, :, 2)$
3: $Blue \leftarrow Image(:, :, 3)$
4: $[m, n] \leftarrow size(Red)$
5: $Combined \leftarrow zeros(m, n \times 3)$
6: **for** $g \leftarrow 1$ to m **do**
7: **for** $h \leftarrow 1$ to n **do**
8: $Combined(g, (h-1) \times 3 + 1) = Red(g, h)$
9: $Combined(g, (h-1) \times 3 + 2) = Green(g, h)$
10: $Combined(g, (h-1) \times 3 + 3) = Blue(g, h)$
11: **end for**
12: **end for**
13: **for** $index \leftarrow 1$ to $3mn$ **do**
14: $length \leftarrow \text{mod}\,(r1(index) + s1(index), 10) + 1$
15: $width \leftarrow \text{mod}\,(r2(index) + s2(index), 10) + 1$
16: **if** $(r1(index) + length) > m \;||\; (s1(index) + width) > n \;||\; (r2(index) + length) > m \;||\; (s2(index) + width) > n$ **then**
17: *continue*
18: **else if** $r2(index) \geq r1(index)$ && $r2(index) \leq (r1(index) + length)$ && $s2(index) \geq s1(index)$ && $s2(index) \leq (s1(index) + width)$ **then**
19: *continue*
20: **else if** $r2(index) \geq r1(index)$ && $s1(index) \leq (r1(index) + length)$ && $(s2(index) + width) \geq s1(index)$ && $(s2(index) + width) \leq (s1(index) + width)$ **then**
21: *continue*
22: **else if** $(r2(index) + length) \geq r1(index)$ && $(s1(index) + length) \leq (r1(index) + length)$ && $(s2(index) + width) \geq s1(k)$ && $(s2(index) + width) \leq (s1(index) + width)$ **then**
23: *continue*
24: **end if**
25: $Rect1 \leftarrow Combined(r1(index) : r1(index) + length, s1(index) : s1(index) + width)$
26: $Rect2 \leftarrow Combined(r2(index) : r2(index) + length, s2(index) : s2(index) + width)$
27: $Combined(r1(index) : r1(index) + length, s1(index) : s1(index) + width) \leftarrow Rect2$
28: $Combined(r2(index) : r2(index) + length, s2(index) : s2(index) + width) \leftarrow Rect1$
29: **end for**
30: $Combined' \leftarrow Combined$
31: *return* $Combined'$

Step 1: Lines 1, 2, and 3 extract the red (Red), green ($Green$), and blue ($Blue$) planes from the given color image $Image$.

Step 2: Line 4 finds the size (m, n) of the color plane Red. Of course, the remaining two color planes would have the same size.

Step 3: On line 5, a new image, $Combined$, is being introduced with all the pixels' values equal to zero and with the size of $(m, n \times 3)$.

Step 4: Lines (6–12) concatenate the three color planes of Red, $Green$, and $Blue$ in a single image $Combined$.

Step 5: The for loop spanning the lines (13–29) carries out the "manufacturing" for the proposed algorithm. Lines (14 and 15) find the length and width of the randomly generated rectangles in the given image. We can see that the streams $r1$, $s1$, $r2$, and $s2$ have been recycled to determine the length and width of the rectangle. In particular, the length and width of each of these rectangles will be a maximum of 10.

Step 6: $if - else - if$ ladder (lines 16–24) checks all the possibilities of two overlapping rectangles. In case these rectangles overlap, the *continue* statement has been fired at lines 17, 19, 21, and 23 which shifts the control back to the for loop header (line 13).

Step 7: In case non-overlapping rectangles are spawned, lines (25–28) swap the pixels $Combined(r1(k) : r1(k) + length, s1(k) : s1(k) + width)$ and $Combined(r2(k) : r2(k) + length, s2(k) : s2(k) + width)$ with each other. We have introduced here two temporary images of $Rect1$ and $Rect2$. They facilitate the proposed swapping process.

Step 8: Line 30 assigns the scrambled image $Combined$ to $Combined'$ and lastly line 31 returns the scrambled image $Combined'$.

Resize the confused and scrambled image $Combined'$ to a new size of $1 \times 3mn$ so that the diffusion process may be done. Now,

carry out the binary operation of Exclusive-OR (XoR) to introduce a diffusion element in the required cipher:

$$Image'(k) = Combined'(k) \oplus mask(k) \tag{3}$$

Here the value of k ranges from 1 to $3mn$ inclusive. Additionally, resize $Image'$ to the size $m \times 3n$. Besides, the algorithm can also be seen in the Figure 3 in the form of a flowchart.

Although both the operations of confusion and diffusion have been successfully carried out, the present cipher image is not a color image *per se*. So, break the image $Image'$ into its constituent components. Further, join these components to have a single encrypted RGB image $Image''$. Of course, the size of the finally obtained image is $m \times n \times 3$, which is compatible with the size of the original input plain image.

To better demonstrate the scrambling process, here an example is given.

Example Figure 4 demonstrates the way, process of scrambling has been carried out in this study. In this particular example, we have taken an image with the size of $6 \times 6 \times 3$. After decomposing the color image into its constituent components/planes and concatenating them horizontally, we get the gray scale image with the size of 6×18. The integers present in this figure correspond to the pixels' data. Figure 4a denotes the starting values of the image. In Figure 4b, we can see that the two randomly formed rectangles are overlapping, so no operation of swapping the pixels encompassed by these two overlapping rectangles has been done. It is to be noted that the colors of the outer boundaries of these two rectangles are blue and red. In the Figure 4c, again the rectangles have been formed randomly. In this case, we can see that they do not overlap with each other. So, Figure 4d shows the given image after swapping the pixels in the non-overlapping rectangles. This is how the scrambling operation has been carried out in this study.

As far as the decryption is concerned, it will be very trivial. The reason for this is that the current study has used the private key cryptography paradigm, so just reversing of the encryption algorithm would render the decryption algorithm.

4 | Simulation of the Suggested Algorithm

Although we have successfully designed a novel color image algorithm in the previous section, this is not sufficient. Here, in this section, the proposed encryption and decryption algorithms will be demonstrated by taking some standard color images. Specifically, four color images—Lena, Lotus, Helicopter, and Octopus— each having a size of 256×256, have been chosen. The images used in the evaluation of the proposed image encryption algorithm were obtained from http://www.vision.caltech.edu/Image_Datasets/Caltech101/, which is a widely recognized and reliable platform for image datasets. These images were selected due to their diversity in content, size, and complexity, ensuring comprehensive testing of the algorithm. The website provides publicly accessible datasets specifically intended for research purposes, ensuring that the images meet ethical and legal standards for use in academic work. Additionally, three color images of Peppers, Sailboat, and Splash and three color images of Airplane, House, and Mandrill of sizes 512×512 and 1024×1024, respectively, have been taken from https://sipi.usc.edu/database/. Apart from that, the MATLAB tool with 64-bit double precision as per IEEE [34] has been used for the experiments.

Figures 5 and 6 display the plaintext images, the ciphertext images, and the decrypted images. It is obvious that our input test images have been transformed into blurred forms in toto, providing no clue or hint of the original images. These phenomena indicate the successful workability of the encryption algorithm. Additionally, the cipher images have been accurately decoded back to their primitive and original forms, validating the effectiveness of the decryption process.

5 | Objective Evaluation of the Suggested Image Cipher

Just the development of image cryptosystems is not sufficient; they must be subjected to varied yardsticks and benchmarks. Over time, cryptographers have introduced an array of metrics for this purpose. In this section, results will be obtained by subjecting the given image cipher against these metrics.

5.1 | Key Space

Brute-force assault is a procedure devised by adversaries in which hackers systematically try all possible secret keys on the encryption algorithm until the correct key is found. The defense against this type of attack is a sufficiently lengthy key space, with a least key space recommended by cryptographers being 2^{100} [35]. In the present task, system parameters as well as initial values work as the key for the suggested image cryptosystem, totaling thirteen (13) key components. The simulation work has been done on the machine whose computer precision is 10^{-14}. So, the key space is found as $10^{14\times13} = 10^{182}$. This extensive key space is highly promising for resisting brute-force attacks. Additionally, Table 1 compares the key space of the suggested cryptosystem with those of other cryptosystems in the field of image security. Our approach surpasses the key space security parameter of the studies presented in [36].

5.2 | Key Sensitivity

Encryption algorithms must be extremely sensitive to the secret key. Therefore, this feature must be carefully evaluated. To show the presence of this feature, a very little alteration is introduced to the secret key of some encryption scheme, and the output is got by sparking the encryption process. The findings must be significantly distinct from those obtained without modifying the secret key.

We assume that $p(0), q(0), r(0), s(0), t(0), a, b, c, d, e, f, g$, and h form the initial key set, referred to as Key_0. The encryption algorithm is applied to an image of Lena (Figure 5a). Next, a very small change of 10^{-14} is made to $p(0)$ of the secret key, resulting in $p(0)' = p(0) + 10^{-14}$. It is important to note that the remaining keys are not altered in this process. This produces another key

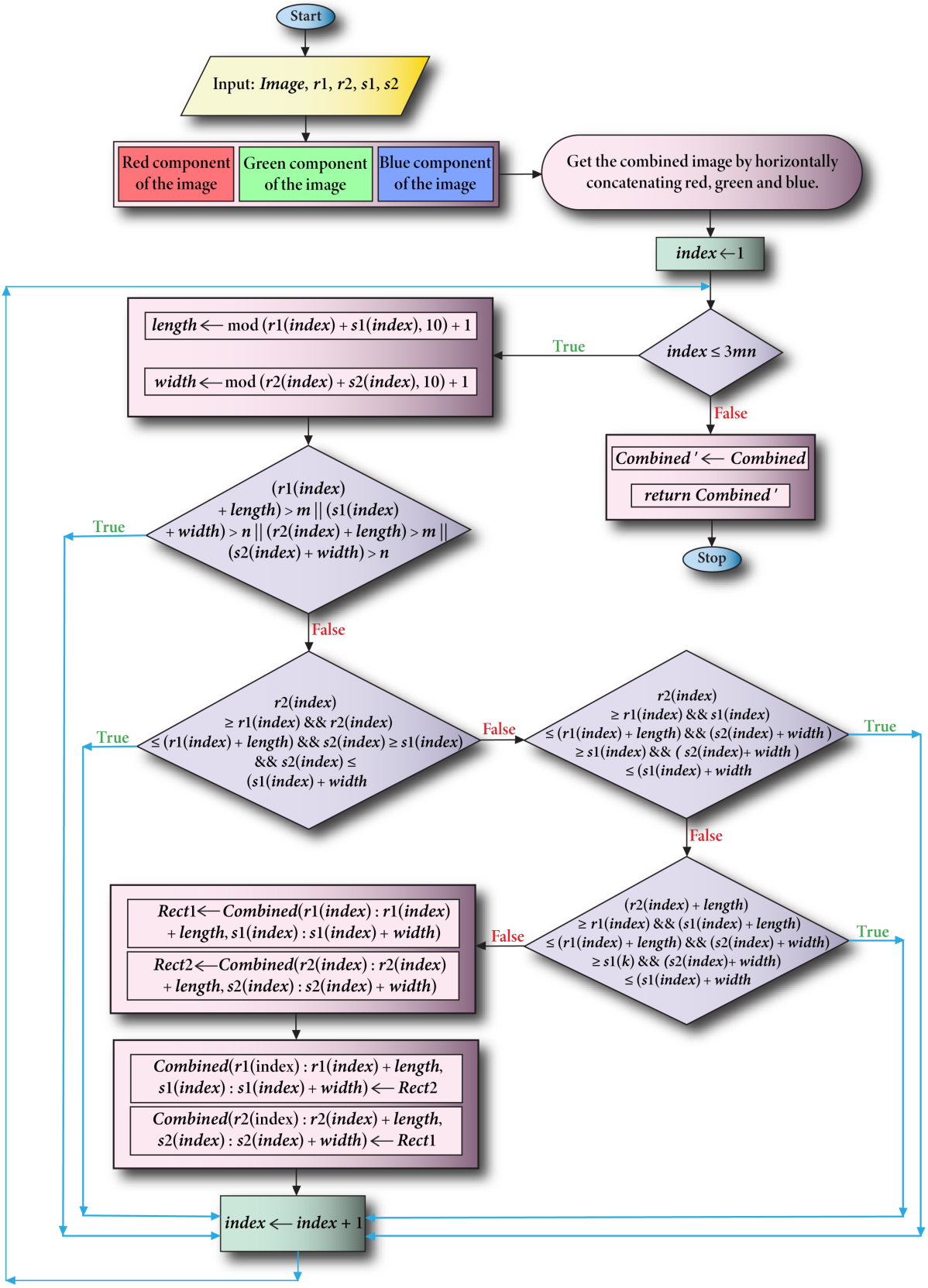
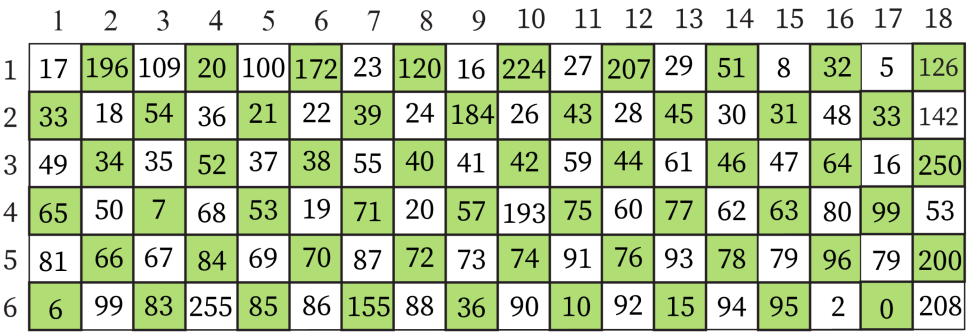
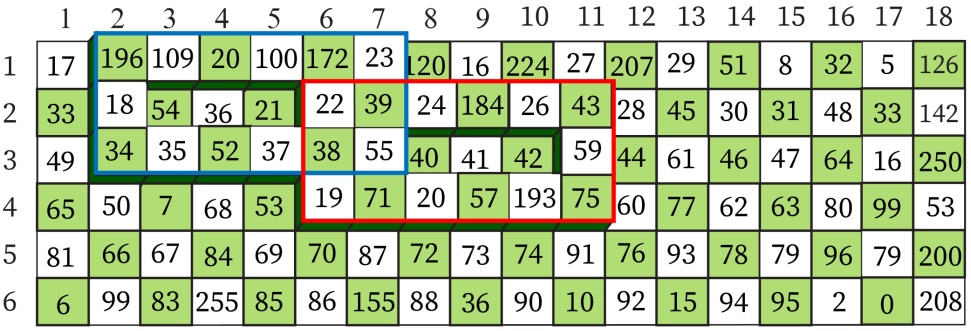


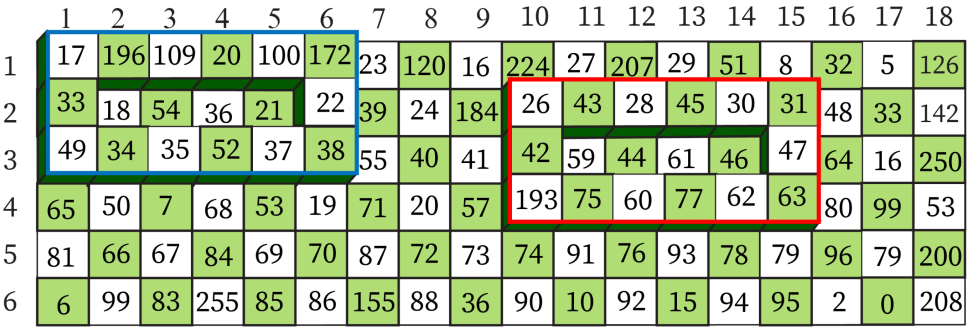
FIGURE 3 | Proposed methodology.



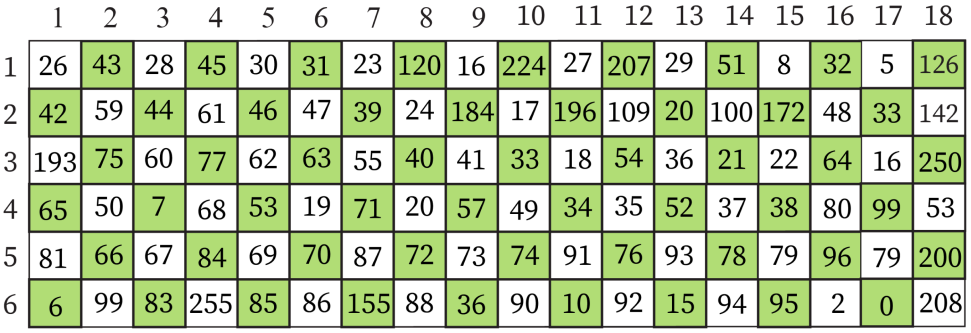
(a)



(b)



(c)



(d)

FIGURE 4 | A demonstration of scrambling an image with the size $6 \times 6 \times 3$: (a) Given image with the size of 6×18; (b) Given image with the overlapping rectangles; (c) Given image with non-overlapping rectangles; (d) Image after swapping the pixels encompassed by the non-overlapping rectangles.

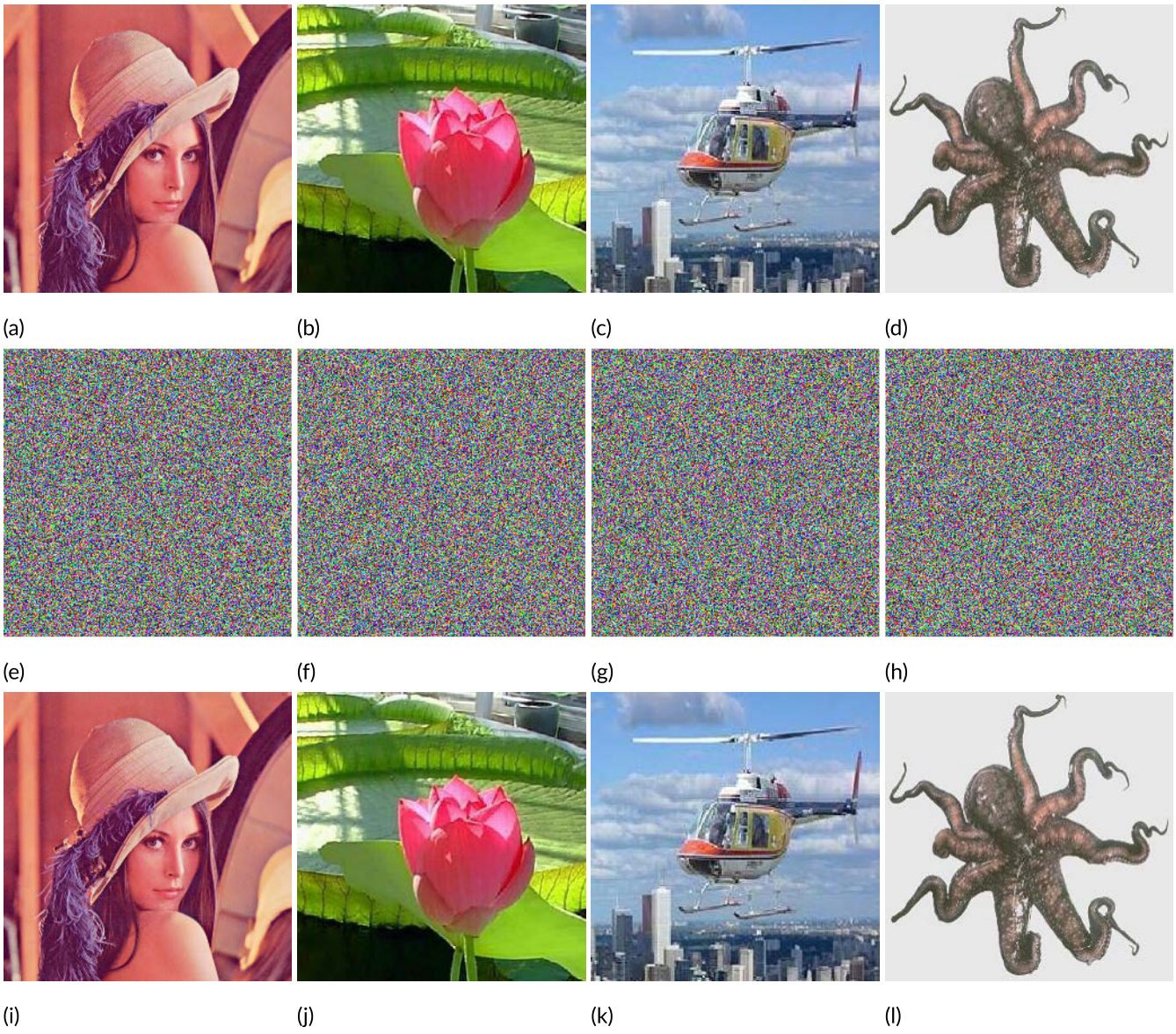


FIGURE 5 | Plain, encrypted, and decrypted images: (a) Lena plain image; (b) Lotus plain image; (c) Helicopter plain image; (d) Octopus plain image; (e) Lena cipher image; (f) Lotus cipher image; (g) Helicopter cipher image; (h) Octopus cipher image; (i) Lena decrypted image; (j) Lotus decrypted image; (k) Helicopter decrypted image; (l) Octopus decrypted image.

set, $p(0)', q(0), r(0), s(0), t(0), a, b, c, d, e, f, g, h$, named Key_1. Using Key_1, the same image of Lena (Figure 5a) is encrypted. The rates of difference (Table 2) are then calculated between the encrypted images produced by Key_0 and $Key_t (t = 1, 2, \ldots, 13)$, where the keys Key_0 and $Key_t (t = 1, 2, \ldots, 13)$ differ slightly. The average rate of difference is 99.61%, which demonstrates that the proposed cryptosystem is highly sensitive to the concerned key. Besides, this result is equal to the results reported in studies [40, 41].

5.3 | Randomness Analysis of the Proposed Algorithm

It is not sufficient to merely produce cipher images by encrypting their plaintext versions using an encryption algorithm. Instead, an objective metric is required to evaluate the intrinsic randomness among the pixels of the cipher images. Fortunately, the NIST Test Suite [42, 43] fulfills this purpose. For the randomness of bit sequences to be accepted, the significance level p for various tests must exceed the threshold of 0.01 [44]. Table 3 presents the results for the four selected images. It can be observed that the cipher images passed all the tests, indicating that their pixel values are sufficiently randomized. This, in turn, suggests robust security effects.

5.4 | Statistical Analysis

Hackers commonly launch statistical attacks against ciphers. In this study, two measures have been selected to demonstrate the robustness of the proposed cipher against such attacks: correlation analysis and histogram analysis.

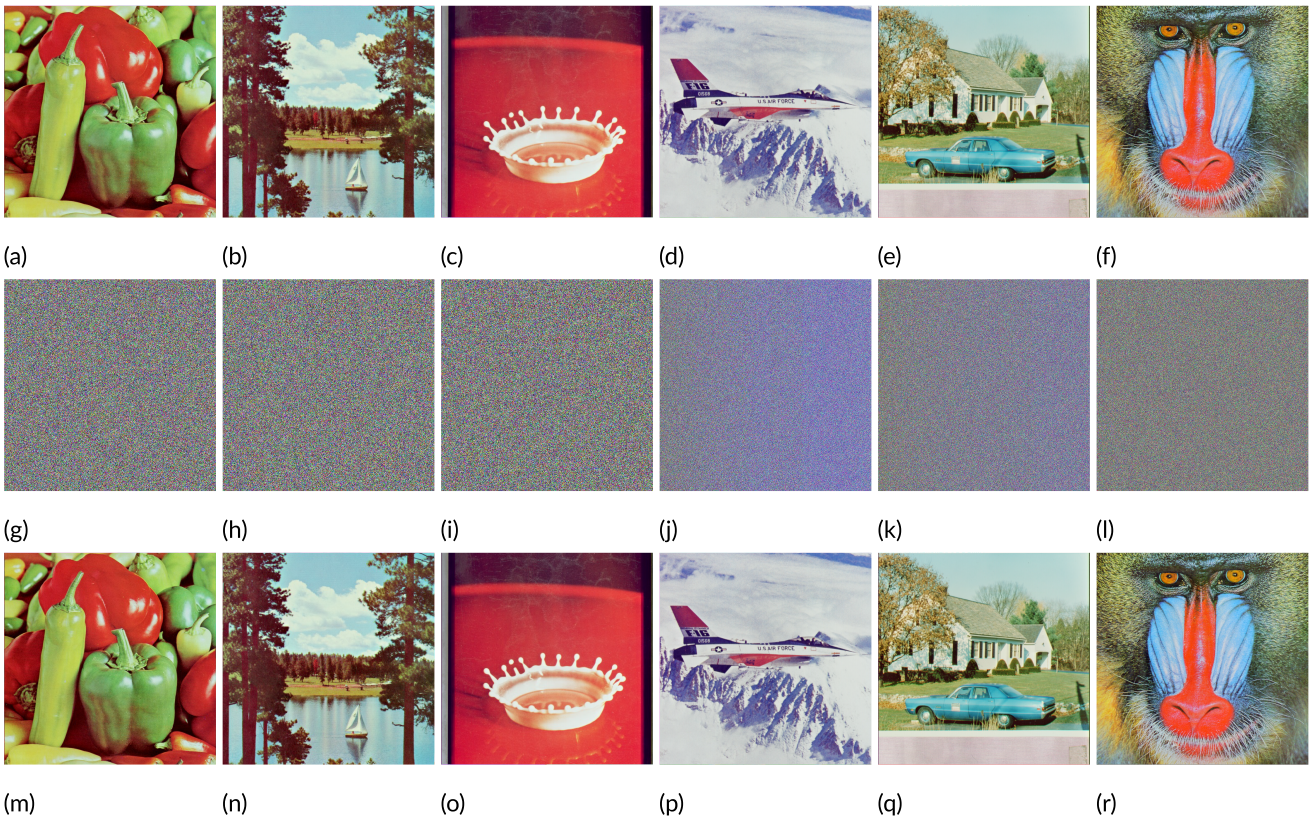


FIGURE 6 | Plain, encrypted, and decrypted images: (a) Peppers plain image; (b) Sailboat plain image; (c) Splash plain image; (d) Airplane plain image; (e) House plain image; (f) Mandrill plain image; (g) Peppers cipher image; (h) Sailboat cipher image; (i) Splash cipher image; (j) Airplane cipher image; (k) House cipher image; (l) Mandrill cipher image; (m) Peppers decrypted image; (n) Sailboat decrypted image; (o) Splash decrypted image; (p) Airplane decrypted image; (q) House decrypted image; (r) Mandrill decrypted image.

TABLE 1 | Key space of present task along comparison.

Work	Key space
Suggested	$10^{14\times13} = 10^{182} \approx 2^{604}$
	$p, q, r, s, t, a, b, c, d, e, f, g, h$
Reference [37]	—
Reference [38]	2^{1754}
Reference [36]	10^{144}
Reference [39]	—

5.4.1 | Analysis of Histogram

The pixels of the images have many intensity values. Due to these intensity values, we get the stunning images. The distribution of pixel intensity values is illustrated using a tool called a histogram. Typically, the histogram of plaintext images is highly irregular, with fluctuating bars. Such histograms attract cryptanalysts because they contain abundant information about the image, making it susceptible to histogram attacks. However, when some image is encoded/encrypted, the resulting histogram of the encoded image has a smooth appearance, making it highly resistant to histogram attacks.

Figures 7 and 8 display histograms for Lena's original and encrypted images in a respective way. This is obvious from these images that the histogram of the plaintext image is irregular and fluctuating, while the encrypted image histogram is more uniform. This uniformity of the histogram serves as a significant impediment against potential assaults of the histogram. Therefore, we posit that the proposed cipher is secure and safe.

Image cryptographers sometimes employ the concept of variance to quantitatively assess variability inherent in the histograms. Lower variance results indicate smoother bars, while higher values indicate greater fluctuations [45–47]. Table 4 presents variance values of histograms for encrypted images of the selected test images. As shown in the table, both the value for Lena image 245.5156 and the average variance value for the selected images 243.6026 are better than the study 264.37 [48]. Therefore, we assert that the proposed method is more secure.

5.4.2 | Chi-Square Test Analysis

The chi-square test is employed to evaluate the uniformity and randomness of the pixel distribution in encrypted images [49]. This statistical test is widely used to verify the strength of encryption algorithms by analyzing whether the frequency of pixel values in the encrypted image adheres to the expected uniform distribution. A uniform distribution ensures that the encryption algorithm generates sufficiently randomized output, reducing the risk of cryptanalysis.

TABLE 2 | Findings of rates of difference: Two images encrypted by minutely distinct keys.

	Difference rates (%)			
Keys employed	**Lena**	**Lotus**	**Helicopter**	**Octopus**
$Key_1(p'(0) = p(0) + 10^{-14})$	99.6068	99.5987	99.6054	99.5823
$Key_2(q'(0) = q(0) + 10^{-14})$	99.6104	99.5962	99.6319	99.6025
$Key_3(r'(0) = r(0) + 10^{-14})$	99.6226	99.5977	99.6203	99.5985
$Key_4(s'(0) = s(0) + 10^{-14})$	99.6155	99.5911	99.6074	99.6254
$Key_5(t'(0) = t(0) + 10^{-14})$	99.6028	99.6251	99.6289	99.6025
$Key_6(a' = a + 10^{-14})$	99.6063	99.6048	99.6059	99.6232
$Key_7(b' = b + 10^{-14})$	99.5758	99.6012	99.6106	99.6282
$Key_8(c' = c + 10^{-14})$	99.6084	99.5906	99.6074	99.6254
$Key_9(d' = d + 10^{-14})$	99.5977	99.6277	99.6258	99.5965
$Key_{10}(e' = e + 10^{-14})$	99.6084	99.6160	99.6254	99.6254
$Key_{11}(f' = f + 10^{-14})$	99.5911	99.6236	99.6158	99.6241
$Key_{12}(g' = g + 10^{-14})$	99.6038	99.5987	99.6254	99.5935
$Key_{13}(h' = g + 10^{-14})$	99.6099	99.6201	99.6157	99.6015
Average	**99.60**	**99.61**	**99.62**	**99.61**
Average of all	**99.61**	—	—	—

TABLE 3 | Randomness analysis results for selected cipher images' p values.

Name	**Lena**	**Lotus**	**Helicopter**	**Octopus**	**Result**
Frequency	0.695412	0.736709	0.605654	0.460954	Pass
Block frequency ($m = 128$)	0.078799	0.054512	0.093987	0.065098	Pass
Cumulative sums (Forward)	0.590765	0.980987	0.800921	0.819954	Pass
Cumulative sums (Reverse)	0.950954	0.763120	0.700567	0.760088	Pass
Runs	0.097654	0.095123	0.270987	0.048712	Pass
Longest run	0.890918	0.609765	0.977611	0.990876	Pass
Rank	0.500299	0.481298	0.807823	0.707723	Pass
FFT	0.709823	0.500879	0.856611	0.870087	Pass
Non overlapping template ($m = 9$, $B = 000000001$)	0.056712	0.097112	0.077908	0.276712	Pass
Overlapping template ($m = 9$)	0.0977812	0.097893	0.023256	0.088873	Pass
Universal	0.288823	0.308922	0.508723	0.707622	Pass
Approximate entropy	0.587234	0.398712	0.589011	0.590762	Pass
Random excursions	0.700993	0.687123	0.509812	0.578254	Pass
Random excursions variant	0.367280	0.490817	0.387201	0.698280	Pass
Serial ($m = 8$)	0.465782	0.898120	0.289761	0.498723	Pass
Linear complexity	0.587210	0.387981	0.487912	0.707725	Pass

The chi-square test statistic, χ^2, is computed using the following equation [50]:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{4}$$

where:

- n is the number of pixel intensity levels (typically 256 for an 8-bit grayscale or color channel).
- O_i represents the observed frequency of pixel intensity i.
- E_i denotes the expected frequency of pixel intensity i.

The null hypothesis (H_0) assumes that the observed pixel frequencies follow a uniform distribution. If the calculated χ^2 value is less than the critical value from the chi-square distribution table at a given significance level (e.g., $\alpha = 0.05$), the null hypothesis cannot be rejected, indicating uniformity in pixel distribution.

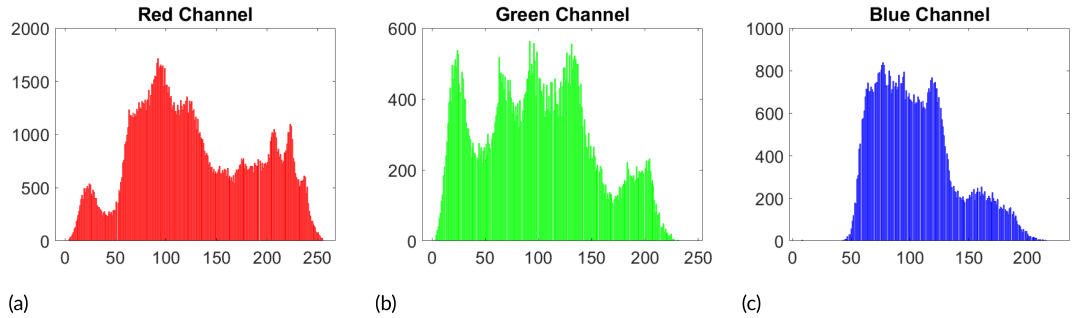


FIGURE 7 | Histograms of standard Lena image in its planes. (a) red plane; (b) green plane; (c) blue plane.

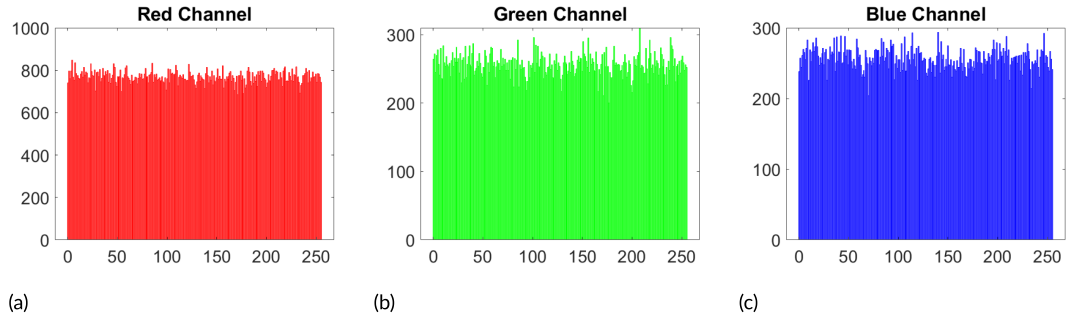


FIGURE 8 | Histogram of cipher Lena image in its planes. (a) red plane, (b) green plane, (c) blue plane.

TABLE 4 | Findings of cipher images' histogram variances.

Study	Lena	Lotus	Helicopter	Octopus	Peppers	Sailboat	Splash	Airplane	House	Mandrill	Average
Suggested	245.5156	234.1875	272.8438	231.6406	243.2776	243.9206	244.7296	241.1483	242.0508	236.7119	**243.6026**
Reference [48]	264.37										

In this study, the chi-square test was applied to the encrypted images produced by the proposed encryption algorithm. Table 5 summarizes the observed and expected frequencies for pixel intensity values. The computed χ^2 statistic was compared against the critical value for $n-1=255$ degrees of freedom at a significance level of 0.05.

The results reveal that the χ^2 statistic falls below the critical threshold, confirming the uniformity of pixel distribution in the encrypted images. This demonstrates that the proposed algorithm generates sufficiently randomized output, ensuring strong resistance against statistical attacks.

Uniform pixel distribution minimizes detectable patterns in the cipher image, making it indistinguishable from random noise. This property is critical for cryptographic security, as any deviation from randomness could provide attackers with exploitable information.

The values of the published works have also been described in the Table 5. According to this table, the results of the proposed work are better than the ones [51, 52] for 256×256-sized images and [50] for the 1024×1024-sized images.

5.4.3 | Analysis of Correlation Coefficient

The pixels in normal and unencrypted images exhibit a strong correlation, resulting in a higher correlation coefficient value. However, when the plain image undergoes encryption, this strong correlation is disrupted, leading to a significant drop in the coefficient value. Typically ranging from −1 to 1, a correlation coefficient value of zero indicates no correlation between consecutive pixels, while a value of 1 or -1 signifies maximal correlation. To illustrate this characteristic of the proposed scheme, pairs of 5,000 randomly selected pixels were analyzed for both the encrypted and unencrypted images. This analysis was conducted in the diagonal, vertical, and horizontal directions using the formula (CC) [25]:

$$CC=\frac{N\sum_{j=1}^{N}(x_j\times y_j)-\sum_{j=1}^{N}x_j\times\sum_{j=1}^{N}y_j}{\sqrt{\left(N\sum_{j=1}^{N}x_j^2-\left(\sum_{j=1}^{N}x_j\right)^2\right)(N\sum_{j=1}^{N}y_j^2-\left(\sum_{j=1}^{N}y_j\right)^2)}} \tag{5}$$

The above equation requires some explanation. In this particular equation, the variable N stands for the number of pixels in some

TABLE 5 | Chi-square test analysis results.

Scheme	Images	Size	Plain			Cipher		
			Red	Green	Blue	Red	Green	Blue
Suggested	Lena	256 × 256	63888.13	28546.01	86487.89	245.51	272.02	232.04
	Lotus	256 × 256	16755.85	20667.71	67928.66	234.18	241.04	289.37
	Helicopter	256 × 256	75834.02	107286.37	88914.10	272.84	265.79	281.99
	Octopus	256 × 256	4289910.50	4307586.14	4181017.71	204.22	261.60	249.53
	Peppers	512 × 512	213187.21	318382.92	491428.17	223.54	219.63	250.07
	Sailboat	512 × 512	196697.30	130154.71	344571.53	271.05	252.23	274.13
	Splash	512 × 512	605662.67	770974.08	1479241.14	234.06	263.96	258.78
	Airplane	1024 × 1024	2679872.99	2679026.06	4384872.99	252.76	261.89	256.87
	House	1024 × 1024	749688.32	1300784.54	970231.49	261.89	248.00	230.56
	Mandrill	1024 × 1024	350438.01	578736.02	328782.50	272.78	267.77	275.12
	Average		924193.50	9721282.14	1024214.45	247.28	255.39	259.84
	Average for all images			5834845.04			254.17	
Reference [51]	Lena	256 × 256		632097.48				958.04
Reference [52]	Lena	256 × 256		634734				980.8
Reference [50]	—	256 × 256						249.7943
		512 × 512						251.9180
		1024 × 1024						261.6742

image. Additionally, the variables x and y are for the pixel color values.

Figure 9 illustrates the correlation distribution of pixels. In particular, three directions of diagonal, horizontal, and vertical have been taken.

The correlation coefficient among neighboring pixels of encrypted and standard images of Lena can be seen in Table 6. According to the table, the metric value is nearly equal to 1 for the original image, whereas it approaches zero for encrypted or cipher images. Figure 9 and Table 6 highlight the significant reduction in the correlation between the pixels of cipher and original images. Furthermore, Table 7 includes a comparative analysis with the state-of-the-art works [36–39]. Undoubtedly, the results of the proposed method are comparable.

5.5 | Entropy Analysis

When a plain image undergoes encryption, its pixels become randomly distributed in the resultant image. In order to quantify their randomness, unpredictability, and disorderliness, the idea of information entropy proves to be quite useful. This concept, introduced by Shannon in 1949 [35], is expressed through the following mathematical formula:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log \frac{1}{p(m_i)} \tag{6}$$

In the equation above, the entropy of the information source m is represented by $H(m)$, where $p(m_i)$ denotes the probability of m_i. The idealized distribution of 256 gray values results in a peak value of 8 for randomized images. Hence, effective encryption algorithms aim to approach this value of 8. The obtained results are presented in Table 8, where the average finding is close to the ideal value of 8. Therefore, it is concluded that this new image cryptosystem is resilient against this type of attack.

According to the Table 8, both the value of Lena and the average value of the 256 × 256 sized images are better than [36]. Besides, the average value of 512 × 512 sized images, 7.9993 is better than [38, 39]. Lastly, the value 7.9994 of the proposed work is better than [53] regarding the 1024 × 1024 sized images.

5.6 | Local Shannon Entropy (LSE)

For laying protection against different attacks, a successful image encryption method should scatter pixels' data of a given image in a random fashion. The notion of local Shannon entropy provides a more precise measure of the randomness of picture pixels [55]. If we consider an image img divided into t non-intersecting blocks $B_1, B_2, \ldots, B_t$, each containing Q_C pixels of data taken randomly, the idea of LSE is defined as follows:

$$\overline{H_{t,Q_C}(img)} = \sum_{i=1}^{t} \frac{H(B_i)}{t} \tag{7}$$

In Equation (7), $H(B_i)$ is for Shannon entropy for image block B_i whose formulation is

$$H(B_i) = \sum_{s=1}^{M} p(s) \log \frac{1}{p(s)} \tag{8}$$

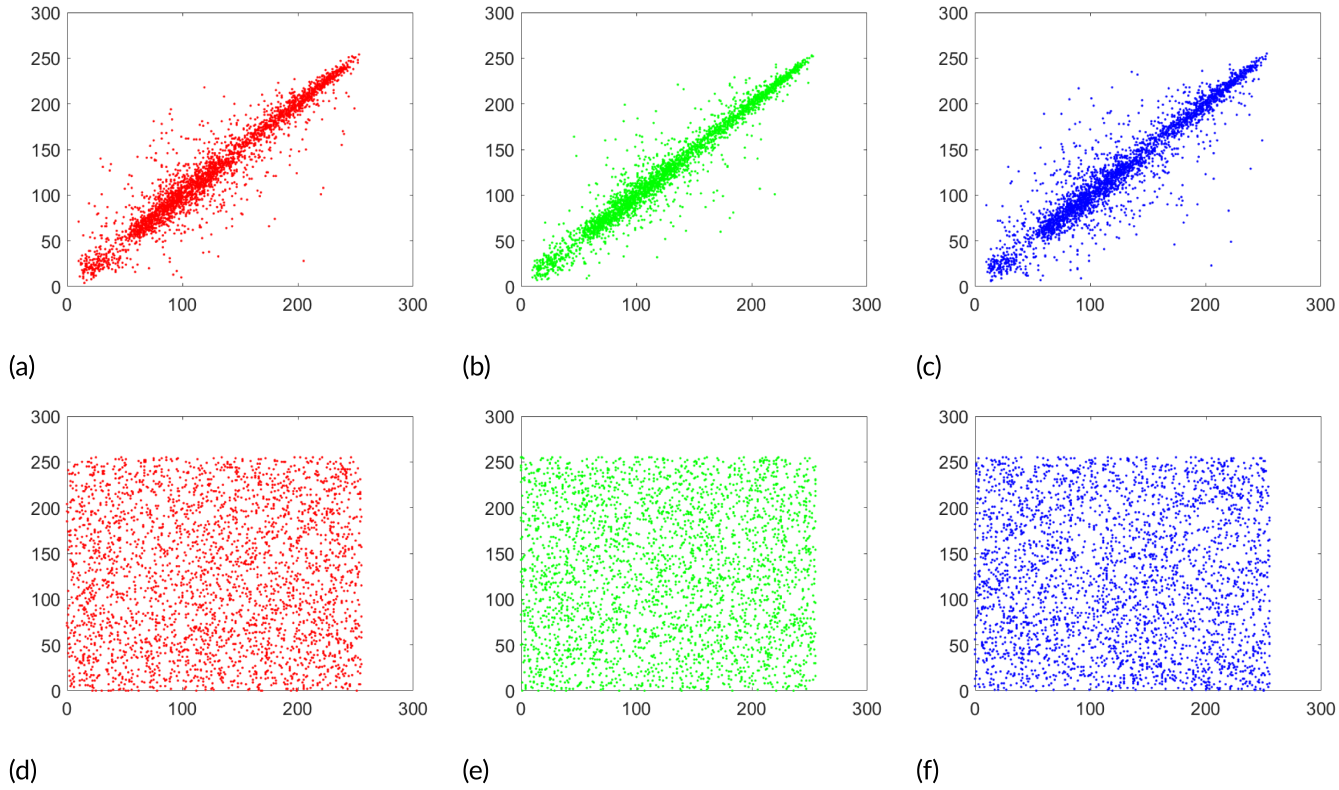


FIGURE 9 | Lena image pixels' correlation distribution: (a) red plane of image, horizontal direction, given standard image; (b) green plane of image, vertical direction, given standard image; (c) blue plane of image, diagonal direction, given standard image; (d) red plane of image, horizontal direction, cipher image; (e) green plane of image, vertical direction, cipher image; (f) blue plane of image, diagonal direction, cipher image.

TABLE 6 | Correlation coefficient findings for plain and cipher images.

		Correlation orientation/direction		
Image	**Plane**	**Horizontal**	**Vertical**	**Diagonal**
Lena (plain image)	Red	0.9432	0.9309	0.9298
	Green	0.9498	0.9312	0.9309
	Blue	0.9312	0.9512	0.9087
Lena (cipher image)	Red	0.0055	0.0034	−0.0032
	Green	0.0087	−0.0026	0.0076
	Blue	−0.0032	0.0012	0.0054

TABLE 7 | Correlation coefficients: Findings of suggested work along with comparison of published works.

		Correlation direction/orientation		
Image	**Work**	**Horizontal**	**Vertical**	**Diagonal**
Standard Lena image		0.9492	0.9321	0.9050
Lena output/cipher image	Ours	−0.0021	0.0058	0.0019
	Reference [37]	0.0156	0.0156	0.0027
	Reference [38]	0.0095	0.0004	0.0006
	Reference [36]	0.0007	0.0031	−0.0051
	Reference [39]	−0.0013	0.0006	0.0018

TABLE 8 | Information entropy results.

Scheme	Images	Size	Plain			Cipher			Average	Average based on size
			Red	Green	Blue	Red	Green	Blue		
Suggested	Lena	256 × 256	7.2507	7.5931	6.9659	7.9973	7.9970	7.9974	**7.9972**	
	Lotus	256 × 256	7.8225	7.7739	7.4316	7.9974	7.9973	7.9968	**7.9972**	
	Helicopter	256 × 256	7.2911	7.1147	7.1857	7.9970	7.9971	7.9969	**7.9970**	
	Octopus	256 × 256	4.4674	4.3920	4.4876	7.9975	7.9975	7.9971	**7.9974**	**7.9972**
	Peppers	512 × 512	7.3388	7.4963	7.0583	7.9994	7.9994	7.9993	**7.9994**	
	Sailboat	512 × 512	7.3124	7.6429	7.2136	7.9993	7.9993	7.9992	**7.9993**	
	Splash	512 × 512	6.9481	6.8845	6.1265	7.9994	7.9993	7.9993	**7.9993**	**7.9993**
	Airplane	1024 × 1024	6.7304	6.8169	6.2256	7.9998	7.9993	7.9995	**7.9995**	
	House	1024 × 1024	7.4242	7.2428	7.4451	7.9998	7.9993	7.9994	**7.9995**	
	Mandrill	1024 × 1024	7.6908	7.4647	7.7445	7.9998	7.9990	7.9992	**7.9993**	**7.9994**
Reference [36]	Lena	256 × 256							7.9966	
Reference [37]	Lena	512 × 512							7.99941	
Reference [38]	Lena	512 × 512							7.999	
Reference [54]	Baboon	512 × 512							7.9994	
Reference [39]	Lena	512 × 512							7.9992	
Reference [50]	Lena	1024 × 1024							7.99985	
Reference [53]	Lena	1024 × 1024							7.999	

Here, M represents the number of pixels, and $p(s)$ denotes the probability of the sth value. Following recommendations outlined in [56], the values assigned to the parameters (t, Q_C) are (30, 1936). Additionally, for $\alpha = 0.05$, the optimal value of LSE is computed as 7.902469317. A cipher image is considered to pass the test if it satisfies the criterion $7.901901305 \leq LSE \leq 7.903037329$. The LSE findings for the selected images are provided in the accompanying Table 9. As one can notice that images successfully passed the LSE test, indicating that the proposed image cryptosystem exhibits desirable chaotic characteristics in the resulting encrypted images.

5.7 | Differential Attack/Plaintext Sensitivity Analysis

At times, adversaries attempt to crack ciphers using a differential attack strategy. In accordance with this method, two versions of a plaintext image are prepared: one as a standard plaintext image and the other one with a minute alteration, such as a change in a single pixel. Subsequently, cipher images are generated for both versions of the plaintext. Through careful analysis of these cipher images, a concealed relationship is uncovered between them, often represented by a mathematical equation. With further analysis, it becomes possible to deduce the key used in the cipher. Cryptographers have developed two measures to counteract this type of attack: (1) Number of Pixels Change Rate ($NPCR$) and (2) Unified Average Changing Intensity ($UACI$). These measures evaluate the alterations observed in the encrypted image when a subtle change is made in the original plain input image. The mathematical formulations of these two measures go as follows:

$$NPCR = \frac{\sum_{u,v} D(u,v)}{M \times N} \times 100\% \tag{9}$$

In the above equation, the pair (M, N) indicates the image's length and width, respectively. Besides, $D(u,v)$ can be expressed as:

$$D(u,v) = \begin{cases} 1, & \text{if } CI(u,v) \neq CI'(u,v) \\ 0, & \text{if } CI(u,v) = CI'(u,v) \end{cases} \tag{10}$$

$$UACI = \frac{1}{M \times N} \left[\sum_{u,v} \frac{|CI(u,v) - CI'(u,v)|}{255} \right] \times 100\% \tag{11}$$

In this equation, CI and CI' represent the cipher images before and after a single pixel of the plaintext image is altered, respectively. The findings of $NPCR$ and $UACI$ for four chosen images are provided in Table 10. To obtain a single value that may represent all the findings, we computed the average results of the metrics of $UACI$ and $NPCR$ across the green, red, and blue channels.

TABLE 9 | Findings of Local Shannon entropy.

Algorithm	Image	Size value	Entropy	Result
Proposed	Lena	256 × 256	7.902376	Passed
	Lotus	256 × 256	7.902902	Passed
	Helicopter	256 × 256	7.902396	Passed
	Octopus	256 × 256	7.902681	Passed
	Peppers	512 × 512	7.902787	Passed
	Sailboat	512 × 512	7.902267	Passed
	Splash	512 × 512	7.902309	Passed
	Airplane	1024 × 1024	7.902678	Passed
	House	1024 × 1024	7.902290	Passed
	Mandrill	1024 × 1024	7.902907	Passed
	Average		7.902559	Passed

TABLE 10 | NPCR and UACI values for different planes and their average values.

			NPCR(%)			UACI(%)		
Algorithm	**Images**	**Size**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Proposed	Lena	256 × 256	99.6089	99.6048	99.6033	33.4569	33.3807	33.5014
	Lotus	256 × 256	99.6078	99.6124	99.6048	33.4918	33.4610	33.5428
	Helicopter	256 × 256	99.6201	99.5758	99.6017	33.5560	33.4956	33.4556
	Octopus	256 × 256	99.5987	99.6231	99.6246	33.4151	33.4082	33.5580
	Peppers	512 × 512	99.6298	99.6098	99.5988	33.4786	33.4287	33.5672
	Sailboat	512 × 512	99.6321	99.6295	99.6098	33.3987	33.4987	33.5677
	Splash	512 × 512	99.6123	99.6098	99.6081	33.4176	33.4365	33.4678
	Airplane	1024 × 1024	99.6412	99.6289	99.5987	33.5678	33.4399	33.5467
	House	1024 × 1024	99.5897	99.6123	99.6354	33.4789	33.4876	33.4876
	Mandrill	1024 × 1024	99.6123	99.5790	99.5276	33.5267	33.4781	33.5123
	Average		**99.6152**	**99.6085**	**99.6012**	**33.4788**	**33.4515**	**33.5207**
	Average for all images			**99.6083**			**33.4836**	
Reference [50]	Lena	256 × 256	99.5895	99.6475	99.5972	33.378	33.4962	33.3907
	Lena	512 × 512	99.6151	99.6071	99.6021	33.4718	33.4148	33.4378
	Lena	1024 × 1024	99.6191	99.5914	99.6140	33.4656	33.4546	33.4508
Reference [57]	—	256 × 256		99.1841			33.5284	
	—	512 × 512		99.6184			33.5739	
	—	1024 × 1024		99.6537			33.6772	

According to the Table 10, the average value of NPCR 99.6083% is better than 99.6081% (the average of 99.6151%, 99.6071%, and 99.6021%) and 99.6082% (the average of 99.6191%, 99.5914%, and 99.6140%) regarding the 512 × 512- and 1024 × 1024-sized images [50]. Additionally, the average value 99.6083% is also better than 99.1841% [57].

As far as the metric UACI is concerned, the average value 33.4836% is better than 33.4216% (the average value of 33.378%, 33.4962%, and 33.3907%) (256 × 256-size image), 33.4416% (the average value of 33.4718%, 33.4148%, and 33.4378%) (512 × 512-size image), and 33.4570% (the average value of 33.4656%, 33.4546%, and 33.4508%) (1024 × 1024-size image) [50].

Tables 11 and 12 present critical values [50] related to the evaluation measures of *UACI* and *NPCR*. For this purpose, images of sizes 256 × 256, 512 × 512, and 1024 × 1024 have been taken. Specifically, critical values $N^*_{0.05}$, $N^*_{0.01}$, and $N^*_{0.001}$ (for three significance levels) are listed in Table 11. These values indicate that if the *NPCR* results for two cipher images are lower than N^*_α, the randomness of these two encrypted images will not be guaranteed at the α significance level. According to Table 13, for the selected

TABLE 11 | NPCR randomness test: Critical values (percentages).

Dimension	$N^*_{0.05}$	$N^*_{0.01}$	$N^*_{0.001}$
256 × 256	99.5693	99.5527	99.5341
512 × 512	99.5893	99.5810	99.5717
1024 × 1024	99.5994	99.5952	99.5906

TABLE 12 | UACI randomness test: Theoretical results (percentages).

Dimension	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$
256 × 256	33.2824	33.7016	33.6777
	33.6447	33.2254	33.1593
512 × 512	33.5541	33.5825	33.6156
	33.3729	33.3445	33.3114
1024 × 1024	33.5088	33.5230	33.5395
	33.4182	33.4040	33.3875

TABLE 13 | NPCR randomness test: Critical values.

Dimension	Image	Result	0.05 level	0.01 level	0.001 level
256 × 256	Lena	99.6154	Pass	Pass	Pass
	Lotus	99.6281	Pass	Pass	Pass
	Helicopter	99.6197	Pass	Pass	Pass
	Octopus	99.6165	Pass	Pass	Pass
512 × 512	Lena	99.6298	Pass	Pass	Pass
1024 × 1024	Lena	99.6271	Pass	Pass	Pass

images encrypted using the proposed encryption scheme, the *NPCR* values for all confidence levels meet the critical (theoretical) threshold for randomness testing across all image sizes of 256 × 256, 512 × 512, and 1024 × 1024.

Regarding the *UACI* parameter, the critical value U^*_α consists of two components, namely U^{*+}_α and U^{*-}_α (as shown in Table 12). The null hypothesis is rejected if the *UACI* value falls outside the interval $(U^{*-}_\alpha, U^{*+}_\alpha)$. Table 14 clearly illustrates that, for images of arbitrary sizes, the *UACI* values adhere to the critical benchmarks set for the *UACI* randomness test.

5.8 | Energy and Contrast Analyses

Serving as another important metric, contrast analysis is used for quantifying intensity variations within an image. Essentially, it evaluates the diversity of pixel intensities in the provided images. Larger results of this metric suggest that images encompass a greater range of gray levels, indicating enhanced security effects. The validation metric for contrast can be defined in mathematical parlance like below [58]:

$$Contrast = \sum_{e,f} |e - f|^2 \times q(e, f) \tag{12}$$

TABLE 14 | UACI randomness test: Critical values.

Dimension	Image	UACI value	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$	$\frac{u^{*-}_{0.05}}{u^{*+}_{0.05}}$
256 × 256	Lena	33.5987	Pass	Pass	Pass
	Lotus	33.4798	Pass	Pass	Pass
	Helicopter	33.5702	Pass	Pass	Pass
	Octopus	33.4987	Pass	Pass	Pass
512 × 512	Lena	33.5175	Pass	Pass	Pass
1024 × 1024	Lena	33.4666	Pass	Pass	Pass

In the context of the given image, (e, f) represents the intensity values, while $q(e, f)$ indicates the frequency at which gray-level co-occurrence matrices (*GLCM*) occur. Putting this in simpler terms, it quantifies how often a pixel with grayscale value e appears in proximity to some other pixel with value of f. Table 15 illustrates the value of this evaluation measure computed using Equation (12) for both the cipher and plain images. The resulting average value of 10.1092 is superior to 8.6448 [58], indicating that the proposed image cipher offers enhanced security.

Besides, an image's energy is written mathematically as follows [58].

$$Energy = \sum_{e,f} q(e, f)^2 \tag{13}$$

In this equation, *Energy* refers to the sum of squared elements in some gray level co-occurrence matrix. additionally, $q(e, f)$ represents the number of gray-level co-occurrence matrices as previously mentioned. This metric serves to quantify the inherent disorder present in the ciphertext image. Lower results in this metric suggest higher encryption quality. Table 16 presents the outcomes of this security parameter for both the encrypted and plain images. It is evident from the table that the values are higher for the plain images, whereas they are lower for the encrypted images. Furthermore, the average value obtained is 0.0188, which surpasses the value of 0.165 reported in [58]. These findings reaffirm the superior security effects achieved by the proposed approach.

5.9 | Peak Signal-to-Noise Ratio Analysis

An exciting project of encryption involves determining the maximum disparity between two things: the ciphertext and plaintext images. To measure this dissimilarity, a similarity metric known as Peak Signal-to-Noise Ratio (PSNR) is employed. Mathematically, it is expressed as [59]:

$$\begin{cases} PSNR = 20\log_{10}(2\max/\sqrt{MSE})dB \\ MSE = \frac{1}{X\times Y\times Z}\sum_{p=1}^{X}\sum_{q=1}^{Y}\sum_{r=1}^{Z}(P_0(p,q,r) - P_1(p,q,r))^2 \end{cases} \tag{14}$$

In this equation, *max* corresponds to the highest scale value of 8 bits grayscale. Besides, Z and X represent dimensions of the given image. Additionally, $P_0(p, q, r)$ and $P_1(p, q, r)$ denote the pixel intensity values of the plaintext and ciphertext images, respectively. Furthermore, MSE denotes the mean squared

TABLE 15 | Findings of contrast analysis.

			Plain image			Cipher image		
Scheme	**Image**	**Size**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Ours	Lena	256 × 256	0.7594	1.0527	0.6394	10.4345	10.4843	10.4098
	Lotus	256 × 256	1.3467	1.8756	1.7612	10.9988	10.0438	10.4553
	Helicopter	256 × 256	0.9541	0.8576	0.7813	10.4113	10.1437	10.4800
	Octopus	256 × 256	1.1324	1.4432	1.2069	10.2435	10.6752	10.7234
	Peppers	512 × 512	0.3250	0.4823	0.3080	10.4484	10.5013	10.4854
	Sailboat	512 × 512	0.4165	0.9717	0.8101	10.5754	10.4860	10.4894
	Splash	512 × 512	0.1350	0.3444	0.3539	10.5076	10.5307	10.4912
	Airplane	1024 × 1024	0.2320	0.2392	0.1715	10.4909	10.4513	6.0603
	House	1024 × 1024	0.2890	0.3399	0.3028	10.5005	10.4666	7.6114
	Mandrill	1024 × 1024	0.7678	0.9892	0.8742	10.4960	10.2623	6.9206
	Average for each component		**0.6357**	**0.8595**	**0.7209**	**10.5106**	**10.4045**	**9.4126**
	Average for all images			**0.7383**			**10.1092**	
Reference [58]	Pepper						8.6448	

TABLE 16 | Findings of energy analysis.

			Plain image			Cipher image		
Work	**Image**	**Size**	**Red**	**Green**	**Blue**	**Red**	**Green**	**Blue**
Ours	Lena	256 × 256	0.1166	0.0699	0.1402	0.0156	0.0156	0.0156
	Lotus	256 × 256	0.1145	0.0444	0.0652	0.0156	0.0156	0.0156
	Helicopter	256 × 256	0.1643	0.1592	0.2519	0.0156	0.0156	0.0156
	Octopus	256 × 256	0.4312	0.0324	0.2340	0.0156	0.0156	0.0156
	Peppers	512 × 512	0.1338	0.1139	0.1769	0.0156	0.0156	0.0156
	Sailboat	512 × 512	0.1079	0.0780	0.1361	0.0156	0.0156	0.0156
	Splash	512 × 512	0.1685	0.1882	0.2466	0.0156	0.0156	0.0156
	Airplane	1024 × 1024	0.3320	0.3531	0.4718	0.0156	0.0157	0.0983
	House	1024 × 1024	0.2890	0.3399	0.3028	0.0156	0.0156	0.0257
	Mandrill	1024 × 1024	0.0644	0.0651	0.0575	0.0156	0.0156	0.0216
	Average for each component		**0.1922**	**0.1444**	**0.2083**	**0.0156**	**0.0156**	**0.0254**
	Average for all images			**0.1816**			**0.0188**	
Reference [58]	Pepper						0.165	

error, which quantifies the amount of disparity between the ciphertext and plaintext images. A higher MSE value results in a lower PSNR value, indicating better security effects.

The PSNR values obtained from various published schemes are presented in Table 17. Notably, the PSNR value is infinite (∞) for the decrypted or restored original plaintext image, signifying that the decrypted image is identical to the original due to $MSE = 0$. This implies that the suggested cryptosystem is lossless. Moreover, in terms of the similarity between the cipher and original plaintext images, the PSNR results for the Lena image using the proposed cipher outperform those of other works [36–39]. Besides, the average value 8.4179 is also better than [36, 37, 39]. Thus, we conclude that the suggested scheme offers superior security effects.

5.10 | Mean Absolute Error (MAE)

Security experts have introduced a large array of parameters to validate their works. MAE is one of them. Mean absolute error is among these metrics, assessing the deviation between two entities: the output cipher image and the input plain image. Its

TABLE 17 | Findings about PSNR: Pair 'P-C' refers to plaintext and cipher images; Pair 'P-D' refers to plaintext and decrypted images.

Size		Lena 256 × 256	Lotus 256 × 256	Helicopter 256 × 256	Octopus 256 × 256	Peppers 512 × 512	Sailboat 512 × 512	Splash 512 × 512	Airplane 1024 × 1024	House 1024 × 1024	Mandrill 1024 × 1024	Average
Suggested	P-D	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	
	P-C	7.8463	8.0479	9.5099	6.9517	9.1185	9.4779	7.5647	8.1608	8.7022	8.7995	**8.4179**
Reference [37]	P-C	9.2399										
Reference [38]	P-C	8.33										
Reference [36]	P-C	8.6512										
Reference [39]	P-C	11.31										

mathematical representation is as follows:

$$MAE_{R,G,B} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |C_{R,G,B}(i,j) - P_{R,G,B}(i,j)| \quad (15)$$

In the given equation, the cipher and plain images are denoted by C and P, respectively, while M and N represent the dimensions of the images. A larger value of MAE indicates better security effects.

Table 18 shows that the average value of 81.8597 for all the chosen images is better than 80.22 [38], 79.6737 (average of 80.0988, 77.8246, and 81.0978), 80.4854 (the average of 80.8166, 78.7709, and 81.8687), 80.8887 (the average of 81.1882, 79.2131, and 82.2648) [50]. Additionally, the value 81.8597 of the proposed work beats all the values of 78.6274, 78.5940, and 77.8732 [60].

5.11 | Noise and Data Loss Threats

In the real world, numerous threats and uncertainties abound. In certain scenarios, cipher images are vulnerable to attacks such as noise and data loss. Cryptosystems that can withstand these attacks are highly esteemed by both academics and practitioners. Therefore, it is imperative for robust image encryption schemes to address these challenges. Noise may be introduced into cipher images during transmission or storage, compromising their integrity. For showing the resilience of the suggested image cryptosystem against the assault of noise, densities of 0.1 and 0.2 have been artificially mixed with the Lena and Lotus cipher images (Figure 10a,b respectively). As we sparked the decryption machinery for these cipher images polluted with noise densities, the results obtained have been drawn in the Figure 10c,d. One can easily appreciate that the information borne by these images is intact.

In real-world scenarios, data loss or crop attacks are occasionally encountered. When cipher images are transmitted to recipients, they may be subject to data loss, leading to potential issues during decryption. Figure 11a,b illustrate cipher images of Helicopter and Octopus with data cropping. In particular, both of these cipher images have been damaged by 50% horizontally and vertically. To demonstrate the efficacy of the suggested cipher in mitigating this attack, these cropped images were subjected to the decryption process. The resulting decoded images are presented in Figure 11c,d. It's evident from these figures that the original cipher images remain intact. This phenomenon, in turn, posits that the proposed image cryptosystem is resilient enough to defy potential crop attacks.

5.12 | Computational Time Analysis

Ensuring security remains a paramount concern when developing cryptographic solutions. Ciphers with shorter response times are often favored for real-world applications. This project was conducted on an Intel(R) Core(TM) i5-4210U CPU @ 1.70 GHz 2.40 GHz processor with 8 GB of memory, utilizing MATLAB R2016a on the Windows 10 operating system.

TABLE 18 | Findings of *MAE*.

Work	Image	Size	MAE		
			Red	Green	Blue
Suggested	Lena	256 × 256	84.3569	78.0275	70.4091
	Lotus	256 × 256	82.4323	79.4495	90.8085
	Helicopter	256 × 256	71.1635	73.8414	85.1874
	Octopus	256 × 256	93.9347	93.8361	94.0937
	Peppers	512 × 512	73.7744	86.7269	86.2217
	Sailboat	512 × 512	71.3780	87.7129	87.8225
	Splash	512 × 512	87.1585	90.8171	81.3904
	Airplane	1024 × 1024	81.4439	84.0535	78.7654
	House	1024 × 1024	76.8917	79.6285	75.9987
	Mandrill	1024 × 1024	76.1356	71.7922	80.5412
	Average for each component		**79.8669**	**82.5885**	**83.1238**
	Average for all images			**81.8597**	
Reference [37]	Lena	256 × 256		85.4321	
Reference [38]	Lena	256 × 256		80.22	
Reference [50]	Lena	256 × 256	80.0988	77.8246	81.0978
	Lena	512 × 512	80.8166	78.7709	81.8687
	Lena	1024 × 1024	81.1882	79.2131	82.2648
Reference [60]	Lena	256 × 256		78.6274	
	Lena	512 × 512		78.5940	
	Lena	1024 × 1024		77.8732	

FIGURE 10 | Aversion of Pepper & Salt noise assault: (a) Cipher Lena image with artificially contaminated 0.1 noise density; (b) Cipher Lotus image with artificially contaminated 0.2 noise density; (c) Decrypted image from (a); (d) Decrypted image from (b).

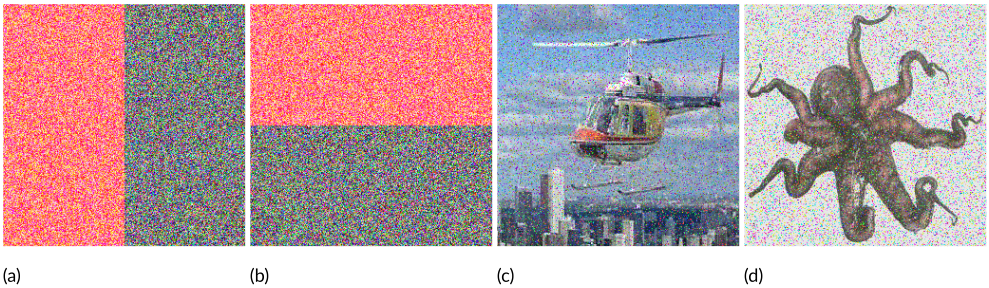


FIGURE 11 | Aversion of data loss attack: (a) 256 × 128 data loss of Helicopter cipher image; (b) 128 × 256 data loss of Octopus cipher image; (c) Retrieved Helicopter image from (a); (d) Retrieved Octopus image from (b).

TABLE 19 | Computational speed of proposed scheme and comparison.

Work	Image	Size	Speed (sec)	Average
Proposed	Lena	256 × 256	1.5025	
	Lotus	256 × 256	1.6254	
	Helicopter	256 × 256	1.6433	
	Octopus	256 × 256	1.5687	**1.5849**
	Peppers	512 × 512	2.8765	
	Sailboat	512 × 512	2.1276	
	Splash	512 × 512	2.7712	**2.5917**
	Airplane	1024 × 1024	3.3088	
	House	1024 × 1024	3.1287	
	Mandrill	1024 × 1024	3.6512	**3.3629**
	Average		**2.4203**	
Reference [61]	Lena	256 × 256	2.5607	
Reference [62]	Lena	256 × 256	3.1143	
Reference [64]	—		0.067230	
Reference [50]	Lena	256 × 256	0.3334	
	Lena	512 × 512	0.9347	
	Lena	1024 × 1024	3.2615	
Reference [63]	—	256 × 256	1.48	
	—	512 × 512	4.81	
	—	1024 × 1024	15.73	
Reference [54]	Baboon	512 × 512	21.044414	

Table 19 provides the results of the proposed work regarding the speed of the encryption algorithm. The average speed of all the 256 × 256 sized images is 1.5849, which is better than 2.5607 [61], 3.1143 [62]. Moreover, the average speed for the 512 × 512 sized images is 2.5917, which beats 4.81 [63] and 21.044414 [54]. Additionally, 3.3629 came out to be the average speed of 1024 × 1024 sized images, which is better than 15.73 [63].

6 | Conclusion

In conclusion, the current research presents a novel image encryption scheme tailored to address the security challenges posed by the proliferation of Internet of Things (IoT) devices. By leveraging chaotic systems and advanced cryptographic techniques, the proposed algorithm effectively encrypts images transmitted and stored within IoT ecosystems, thereby mitigating the risk of privacy and security breaches. Through rigorous testing on standard benchmark images, our algorithm demonstrates significant improvements in security metrics such as entropy, correlation coefficient, and histogram uniformity. Furthermore, its robustness against common cryptographic attacks, including differential and statistical attacks, has been thoroughly evaluated, highlighting its effectiveness in safeguarding image data against unauthorized access. Importantly, the algorithm's performance in terms of computational efficiency has been assessed, revealing low computational overhead suitable for resource-constrained IoT devices. This attribute enhances its practicality and applicability in real-world IoT scenarios. Overall, the findings of this research contribute to the advancement of secure IoT applications, ensuring the integrity and confidentiality of visual data in an increasingly interconnected world. By addressing the critical concern of image security in IoT ecosystems, our work lays the foundation for enhanced data protection and privacy in the era of pervasive connectivity. In future, the authors would like explore novel technologies such as machine learning and blockchain [65–70] for secure data transmission in IoT ecosystem.

Author Contributions

Muhammad Aqeel: methodology; conceptualization; writing – original draft. **Arfan Jaffar:** project administration; writing – review and editing; supervision. **Muhammad Faheem:** software; data validation; review and editing. **Muhammad Waqar Ashraf:** validation; investigation. **Nadeem Iqbal:** writing – review and editing; methodology. **Shahid Yousaf:** visualization; writing – review and editing; software. **Hossam Diab:** investigation; writing – review and editing.

Acknowledgments

The authors would like to thank their affiliated universities and institutes for supporting this research work.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data will be available upon request to the corresponding author.

Peer Review

The peer review history for this article is available at https://publons.com/publon/10.1002/eng2.13099.

References

1. K. Chen, B. Chen, C. Liu, W. Li, Z. Zou, and Z. Shi, "RSMAMBA: Remote Sensing Image Classification With State Space Model," *IEEE Geoscience and Remote Sensing Letters* 19, no. 8 (2024): 10821.

2. R. Pandey, S. Saha, N. Yathiraju, I. S. Abdulrahman, R. Nittala, and V. Tripathi, "Integration of RFID and Image Processing for Surveillance Abased Security System," in *3rd International Conference on Advance Computing and Innovative Technologies in Engineering* (Barcelona, Spain: IEEE, 2023), 380–384.

3. S. Zhang and D. Metaxas, "On the Challenges and Perspectives of Foundation Models for Medical Image Analysis," *Medical Image Analysis* 12, no. 8 (2023): 102996.

4. A. Al-Habaibeh, S. Yaseen, and B. Nweke, "A Comparative Study of Low and High Resolution Infrared Cameras for IoT Smart City Applications," *Ain Shams Engineering Journal* 14, no. 6 (2023): 102108.

5. B. Zhang and L. Liu, "Chaos-Based Image Encryption: Review, Application, and Challenges," *Mathematics* 11, no. 11 (2023): 2585.

6. F. L. Færøy, M. M. Yamin, A. Shukla, and B. Katt, "Automatic Verification and Execution of Cyber Attack on Iot Devices," *Sensors* 23, no. 2 (2023): 733.

7. S. O. Olabanji, O. B. Oladoyinbo, C. U. Asonze, Oladoyinbo TO, S. A. Ajayi, and O. O. Olaniyi, "Effect of Adopting AI to Explore Big Data on Personally Identifiable Information (PII) for Financial and Economic Data Transformation," 2024 Available at SSRN 4739227.

8. Y. Chen and P. Esmaeilzadeh, "Generative AI in Medical Practice: In-Depth Exploration of Privacy and Security Challenges," *Journal of Medical Internet Research* 26 (2024): e53008.

9. S. Subaselvi, C. Mytheesh, R. Sanjay, S. Ragunath, et al., "VLSI Implementation of Triple-DES Block Cipher," in *In: 2023 7th International Conference on Computing Methodologies and Communication (ICCMC)* (Erode, India: IEEE, 2023), 1162–1166.

10. M. Santhanalakshmi, M. Lakshana, and G. M. Shahitya, "Enhanced AES-256 Cipher Round Algorithm for IoT Applications," *Scientific Temper* 14, no. 1 (2023): 184–190.

11. L. Matysiak, "Generalized RSA Cipher and Diffie-Hellman Protocol," *Journal of Applied Mathematics and Informatics* 39, no. 1_2 (2021): 93–103.

12. R. Robet, O. Pribadi, S. Widiono, M. K. Sarker, et al., "Image Encryption Using Half-Inverted Cascading Chaos Cipheration," *Journal of Computing Theories and Applications* 1, no. 2 (2023): 61–77.

13. P. N. Andono, et al., "Improved Pixel and Bit Confusion-Diffusion Based on Mixed Chaos and Hash Operation for Image Encryption," *IEEE Access* 10 (2022): 115143–115156.

14. E. Winarno, K. Nugroho, P. W. Adi, et al., "Combined Interleaved Pattern to Improve Confusion-Diffusion Image Encryption Based on Hyperchaotic System," *IEEE Access* 11 (2023): 69005–69021.

15. R. Dehghani and H. Kheiri, "Chaotic-Based Color Image Encryption Using a Hybrid Method of Reversible Cellular Automata and DNA Sequences," *Multimedia Tools and Applications* 83, no. 6 (2024): 17429–17450.

16. A. Y. Darani, Y. K. Yengejeh, H. Pakmanesh, and G. Navarro, "Image Encryption Algorithm Based on a New 3D Chaotic System Using Cellular Automata," *Chaos, Solitons & Fractals* 179 (2024): 114396.

17. K. Kumar, S. Roy, U. Rawat, and A. Shandilya, "SOCIET: Second-Order Cellular Automata and Chaotic Map-Based Hybrid Image Encryption Technique," *Multimedia Tools and Applications* 83, no. 10 (2024): 29455–29484.

18. Y. Cui, J. Guo, C. Shang, et al., "Light-Field 3D Image Parallel Encryption Based on the State Transition Diagram of Maximum Length Cellular Automata," *Optics Communications* 552 (2024): 130063.

19. Y. Chen, H. Huang, C. Tang, and W. Wei, "A Novel Adaptive Image Privacy Protection Method Based on Latin Square," *Nonlinear Dynamics* 22 (2024): 1–24.

20. R. Huang, H. Liu, X. Liao, and A. Dong, "On the Cryptanalysis of a Latin Cubes-Based Image Cryptosystem," *Entropy* 23, no. 2 (2021): 202.

21. M. U. Hassan, A. Alzayed, A. A. Al-Awady, N. Iqbal, M. Akram, and A. Ikram, "A Novel RGB Image Obfuscation Technique Using Dynamically Generated all Order-4 Magic Squares," *IEEE Access* 11 (2023): 46382–46398.

22. A. Renklier and S. Öztürk, "Image Authentication and Recovery: Sudoku Puzzle and MD5 Hash Algorithm Based Self-Embedding Fragile Image Watermarking Method," *Multimedia Tools and Applications* 83, no. 5 (2024): 13929–13951.

23. K. D. S. Kiran, K. Bharath, J. Yashwanth, B. N. Patel, and K. Prabhavathi, "Optimized Encryption Technique for Securing E-Health Images," in *International Conference on Computer Vision and Robotics* (Wuxi, China: Springer, 2023), 131–141.

24. A. A. Neamah, "An Image Encryption Scheme Based on a Seven-Dimensional Hyperchaotic System and Pascal's Matrix," *Journal of King Saud University, Computer and Information Sciences* 35, no. 3 (2023): 238–248.

25. M. M. Hazzazi, N. Iqbal, and A. Ikram, "Digital Images Security Technique Using Hénon and Piecewise Linear Chaotic Maps," *IEEE Access* 22, no. 8 (2023): 12721.

26. J. Sun, "2D-SCMCI Hyperchaotic Map for Image Encryption Algorithm," *IEEE Access* 9 (2021): 59313–59327.

27. Q. Jiang, S. Yu, and Q. Wang, "Cryptanalysis of an Image Encryption Algorithm Based on Two-Dimensional Hyperchaotic Map," *Entropy* 25, no. 3 (2023): 395.

28. M. Alshehri, S. Almakdi, M. Al Qathrady, and J. Ahmad, "Cryptanalysis of 2D-SCMCI Hyperchaotic Map Based Image Encryption Algorithm," *Computer Systems Science and Engineering* 46, no. 2 (2023): 2401–2414.

29. X. Gao, "Image Encryption Algorithm Based on 2D Hyperchaotic Map," *Optics & Laser Technology* 142 (2021): 107252.

30. D. Ding, W. Wang, Z. Yang, et al., "An n-Dimensional Modulo Chaotic System With Expected Lyapunov Exponents and Its Application in Image Encryption," *Chaos, Solitons & Fractals* 174 (2023): 113841.

31. H. Natiq, A. Roy, S. Banerjee, A. Misra, and N. Fataf, "Enhancing Chaos in Multistability Regions of Duffing Map for an Image Encryption Algorithm," *Soft Computing* 27, no. 24 (2023): 19025–19043.

32. J. Alqahtani, M. Akram, G. A. Ali, N. Iqbal, A. Alqahtani, and R. Alroobaea, "Elevating Network Security: A Novel S-Box Algorithm for Robust Data Encryption," *IEEE Access* 24, no. 10 (2023): 13838.

33. Y. Li, C. Wang, and H. Chen, "A Hyper-Chaos-Based Image Encryption Algorithm Using Pixel-Level Permutation and Bit-Level Permutation," *Optics and Lasers in Engineering* 90 (2017): 238–246.

34. W. Mao, K. Li, Q. Cheng, et al., "A Configurable Floating-Point Multiple-Precision Processing Element for HPC and AI Converged Computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 30, no. 2 (2021): 213–226.

35. N. Iqbal, S. Abbas, M. A. Khan, T. Alyas, A. Fatima, and A. Ahmad, "An RGB Image Cipher Using Chaotic Systems, 15-Puzzle Problem and DNA Computing," *IEEE Access* 7 (2019): 174051–174071.

36. W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, and A. Aboshousha, "Color Image Encryption Through Chaos and Kaa Map," *IEEE Access* 11 (2023): 11541–11554.

37. E. T. Oladipupo, O. C. Abikoye, and J. B. Awotunde, "A Lightweight Image Cryptosystem for Cloud-Assisted Internet of Things," *Applied Sciences* 14, no. 7 (2024): 2808.

38. W. Alexan, D. El-Damak, and M. 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* 8, no. 6 (2024): 1–18.

39. S. O. Hwang, H. M. Waseem, and N. Munir, "Billiard Quantum Chaos: A Pioneering Image Encryption Scheme in the Post-Quantum Era," *IEEE Access* 9, no. 8 (2024): 426–436.

40. A. Kulsoom, D. Xiao, and S. A. Abbas, "An Efficient and Noise Resistive Selective Image Encryption Scheme for Gray Images Based on Chaotic Maps and DNA Complementary Rules," *Multimedia Tools and Applications* 75 (2016): 1–23.

41. X. Liao, M. A. Hahsmi, R. Haider, et al., "An Efficient Mixed Inter-Intra Pixels Substitution at 2bits-Level for Image Encryption Technique Using DNA and Chaos," *Optik-International Journal for Light and Electron Optics* 153 (2018): 117–134.

42. S. Gao, R. Wu, X. Wang, et al., "A 3D Model Encryption Scheme Based on a Cascaded Chaotic System," *Signal Processing* 202 (2023): 108745.

43. R. Wu, S. Gao, X. Wang, et al., "AEA-NCS: An Audio Encryption Algorithm Based on a Nested Chaotic System," *Chaos, Solitons & Fractals* 165 (2022): 112770.

44. E. Yavuz, R. Yazici, M. C. Kasapbaşi, and E. Yamaç, "A Chaos-Based Image Encryption Algorithm With Simple Logical Functions," *Computers and Electrical Engineering* 54 (2016): 471–483.

45. D. S. Malik and T. Shah, "Color Multiple Image Encryption Scheme Based on 3D-Chaotic Maps," *Mathematics and Computers in Simulation* 178 (2020): 646–666.

46. J. I. M. Bezerra, G. Machado, A. Molter, R. I. Soares, and V. Camargo, "A Novel Simultaneous Permutation–Diffusion Image Encryption Scheme Based on a Discrete Space Map," *Chaos, Solitons & Fractals* 168 (2023): 113160.

47. A. Durdu, "Image Transfer With Secure Communications Application Using a New Reversible Chaotic Image Encryption," *Multimedia Tools and Applications* 83, no. 2 (2024): 3397–3424.

48. X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A Color Image Cryptosystem Based on Dynamic DNA Encryption and Chaos," *Signal Processing* 155 (2019): 44–62.

49. H. Khanzadi, M. Eshghi, and S. E. Borujeni, "Image Encryption Using Random Bit Sequence Based on Chaotic Maps," *Arabian Journal for Science and Engineering* 39 (2014): 1039–1047.

50. M. Hanif, S. Abbas, M. A. Khan, et al., "A Novel and Efficient Multiple RGB Images Cipher Based on Chaotic System and Circular Shift Operations," *IEEE Access* 8 (2020): 146408–146427.

51. P. Parida, C. Pradhan, X. Z. Gao, D. S. Roy, and R. K. Barik, "Image Encryption and Authentication With Elliptic Curve Cryptography and Multidimensional Chaotic Maps," *IEEE Access* 9 (2021): 76191–76204.

52. Y. Luo, X. Ouyang, J. Liu, and L. Cao, "An Image Encryption Method Based on Elliptic Curve Elgamal Encryption and Chaotic Systems," *IEEE Access* 7 (2019): 38507–38522.

53. R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem, and M. Lee, "Image Encryption Using a Synchronous Permutation-Diffusion Technique," *Optics and Lasers in Engineering* 90 (2017): 146–154.

54. Z. Xiong, Y. Wu, C. Ye, X. Zhang, and F. Xu, "Color Image Chaos Encryption Algorithm Combining CRC and Nine Palace Map," *Multimedia Tools and Applications* 78, no. 22 (2019): 31035–31055.

55. B. Rezaei, M. Mobasseri, and R. Enayatifar, "A Secure, Efficient and Super-Fast Chaos-Based Image Encryption Algorithm for Real-Time Applications," *Journal of Real-Time Image Processing* 20, no. 2 (2023): 30.

56. D. H. ElKamchouchi, H. G. Mohamed, and K. H. Moussa, "A Bijective Image Encryption System Based on Hybrid Chaotic Map Diffusion and DNA Confusion," *Entropy* 22, no. 2 (2020): 180.

57. Z. Shao, X. Liu, Q. Yao, N. Qi, Y. Shang, and J. Zhang, "Multiple-Image Encryption Based on Chaotic Phase Mask and Equal Modulus Decomposition in Quaternion Gyrator Domain," *Signal Processing: Image Communication* 80 (2020): 115662.

58. F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, and M. A. Khan, "A Novel Image Encryption Based on Lorenz Equation, Gingerbreadman Chaotic Map and S 8 Permutation," *Journal of Intelligent & Fuzzy Systems* 33, no. 6 (2017): 3753–3765.

59. D. R. I. M. Setiadi, "PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography," *Multimedia Tools and Applications* 80, no. 6 (2021): 8423–8444.

60. M. Hanif, R. A. Naqvi, S. Abbas, M. A. Khan, and N. Iqbal, "A Novel and Efficient 3D Multiple Images Encryption Scheme Based on Chaotic Systems and Swapping Operations," *IEEE Access* 8 (2020): 123536–123555.

61. N. Iqbal, M. Hanif, Z. U. Rehman, and M. Zohaib, "On the Novel Image Encryption Based on Chaotic System and DNA Computing," *Multimedia Tools and Applications* 81, no. 6 (2022): 8107–8137.

62. Z. Bashir, N. Iqbal, and M. Hanif, "A Novel Gray Scale Image Encryption Scheme Based on Pixels' Swapping Operations," *Multimedia Tools and Applications* 80 (2021): 1029–1054.

63. X. Liao, A. Kulsoom, and S. Ullah, "A Modified (Dual) Fusion Technique for Image Encryption Using SHA-256 Hash and Multiple Chaotic Maps," *Multimedia Tools and Applications* 75, no. 18 (2016): 11241–11266.

64. T. Nestor, N. J. De Dieu, K. Jacques, E. J. Yves, A. M. Iliyasu, and A. A. Abd El-Latif, "A Multidimensional Hyperjerk Oscillator: Dynamics Analysis, Analogue and Embedded Systems Implementation, and Its Application as a Cryptosystem," *Sensors* 20, no. 1 (2019): 83.

65. M. Faheem, M. A. Al-Khasawneh, A. A. Khan, and S. H. H. Madni, "Cyberattack Patterns in Blockchain-Based Communication Networks for Distributed renewable Energy Systems: A Study on Big Datasets," *Data in Brief* 53, no. 5 (2024a): 110212, https://doi.org/10.1016/j.dib.2024.11021250.

66. B. Raza, Y. J. Kumar, A. K. Malik, A. Anjum, and M. Faheem, "Performance Prediction and Adaptation for Database Management System Workload Using Case-Based Reasoning Approach," *Information Systems* 76, no. 5 (2018): 46–58, https://doi.org/10.1016/j.is.2018.04.00551.

67. M. Faheem and A.-K. Mahmoud Ahmad, "Multilayer Cyberattacks Identification and Classification Using Machine Learning in Internet of Blockchain(IoBC)-Based Energy Networks," *Data in Brief* 54, no. 5 (2024): 110461, https://doi.org/10.1016/j.dib.2024.110461.

68. M. Faheem, B. Raza, M. S. Bhutta, and S. H. H. Madni, "A Blockchain-Based Resilient and Secure Framework for Events Monitoring and Control in Distributed Renewable Energy Systems," *IET Blockchain* (2024b): 1–15, https://doi.org/10.1049/blc2.12081.

69. A. A. Khan, R. K. Madendran, U. Thirunavukkarasu, and M. Faheem, "D^2PAM: Epileptic Seizures Prediction Using Adversarial Deep Dual Patch Attention Mechanism," *CAAI Transactions on Intelligence Technology.* 8, no. 3 (2023): 755–769, https://doi.org/10.1049/cit2.12261.

70. M. Faheem, M. Umar, R. A. Butt, B. Raza, M. A. Ngadi, and V. C. Gungor, "Software Defined Communication Framework for Smart Grid to Meet Energy Demands in Smart Cities," In *2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)* (2019), 51–55.