A Novel Image Encryption Algorithm Based on Cyclic Chaotic Map in Industrial IoT Environments

Moatsum Alawida, *Member, IEEE*

***Abstract*—In the Industrial Internet of Things (IIoT), ensuring timely and secure data transmission between sensors and edge devices is paramount, particularly when dealing with sensitive information captured by high-resolution image sensors. However, existing methods often struggle to strike the delicate balance between security and efficiency, resulting in either vulnerable transmissions or significant processing delays. This article proposes a novel image encryption algorithm specifically designed for IIoT environments with constrained resources. The proposed algorithm leverages a new chaotic model that combines a cyclic construction with a 1-D perturbed logistic map. The proposed chaos model produces a single data sequence, subsequently utilized to generate three matrices mirroring the size of the image. Among these matrices, two contribute to crafting a permutation matrix for randomizing pixel positions, while the third matrix facilitates diffusion operations. The chaotic data sequence and generated matrices are preprocessed and can be used for encrypting multiple images under the same session key, enhancing efficiency. Encryption utilizes a single round that combines diffusion and permutation operations simultaneously, further reducing processing time. Experimental results demonstrate its effectiveness on an IoT camera sensor for encryption and a separate device for decryption. Statistical tests confirm the robustness of the encrypted images against various attacks, including correlation analysis, histogram analysis, differential attacks, and key and plaintext sensitivity. Furthermore, comparisons with existing image encryption techniques showcase the proposed algorithm's superior security and efficiency. Notably, it effectively encrypts images of varying sizes, making it suitable for deployment in IIoT environments.**

***Index Terms*—Chaotic image cipher, encryption, Industrial Internet of Things (IIoT), IoT camera sensor.**

I. Introduction

THE Industrial Internet of Things (IIoT) has emerged as a pivotal technology in various industrial applications and production processes. IIoT devices, including sensors, controllers, smart devices, and IoT cameras equipped with sensing capabilities, are designed to gather diverse data within industrial environments [1], [2]. Among the types of data collected, digital images stand out due to their significant size and the inclusion of precise and sensitive data. IoT camera sensors, being constrained devices, establish connections with edge devices via wired or wireless means, such as USB or 5G networks. These sensors possess the capability to monitor motion states, capture up to 30 photos per second, transmit captured photos to designated destinations for analysis, and trigger system-wide alerts in critical scenarios.

The imperative for swift data transmission arises from the need for the system to respond rapidly to the captured images. Despite their importance, a notable concern surrounds the security of communication channels employed by most IoT camera sensors [3], [4]. These channels often lack robust security measures, potentially exposing the captured photos to unauthorized access and unauthorized modifications.

Cryptographic solutions play a crucial role in securing captured images from IoT camera sensors [5], aiming for secure communication and storage. Encryption serves as the optimal solution to prevent unauthorized access to stored and transmitted data. However, given the constrained power and memory of IoT camera sensors, traditional encryption methods, such as advanced encryption standard (AES), elliptic curve cryptography (ECC), and others, prove to be time-consuming, with AES taking more than 2 s to encrypt a single image on a standard PC.

On another side, lightweight encryption algorithms have been proposed, emphasizing a reduction in the size of the secret key and the number of encryption rounds. This approach gains significance when dealing with small datasets, such as temperature readings, raw data, simple numerical values, industrial control systems, and device statuses. Nevertheless, lightweight encryption algorithms also require time for encrypting captured images, relying on the encryption of blocks over multiple rounds to achieve the desired security level. To encrypt images captured by IoT camera sensors, it is imperative to maintain a secret key size exceeding 128, aligning with standard cybersecurity practices globally. Simultaneously, reducing the number of rounds without compromising the high-security levels required is essential.

Indeed, image encryption presents distinct challenges compared with text encryption. Unlike text data, images typically exhibit high spatial correlation, where adjacent pixels tend to

Manuscript received 24 January 2024; revised 11 March 2024, 22 March 2024, and 3 April 2024; accepted 10 April 2024. Date of publication 13 May 2024; date of current version 5 August 2024. This work was supported by Abu Dhabi University under Grant 19300786. Paper no. TII-24-0379.

The author is with the Department of Computer Sciences and Information Technology, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates (e-mail: moatsum.alawida@adu.ac.ae).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3395631.

Digital Object Identifier 10.1109/TII.2024.3395631

share similar values [6]. This characteristic poses a hurdle for traditional block-based encryption algorithms, such as AES [5], [7], particularly when applied to large images, as they may not adequately address this correlation. Ideally, image encryption techniques should treat the entire image holistically [8], rather than processing it in blocks, to ensure both faster processing and enhanced security against the potential exploitation of these correlations.

During the past decades, chaotic maps have garnered significant attention from researchers due to their foundational characteristics [9], including properties, such as initial condition sensitivity, random number generation, unpredictability, and ergodicity. In this context, Sha et al. [10] proposed a flexible and secure chaotic image cryptosystem designed for encrypting various sizes and types of images, including grayscale, color, and 3-D images within the IIoT domain. Their approach involved employing a graph operation on the image, extracting a flexible matrix, and using zigzag to scramble the image. Nonsequential diffusion was then applied between interblock and intrablock pixels. Notably, the algorithm was designed to operate in a single round to achieve a high level of security. However, it is essential to acknowledge that the proposed algorithm has a lengthy secret key, and the chaotic maps utilized involve multiple mathematical operations.

Gu et al. [11] proposed a 16-bit parallel chaotic system designed for encrypting images in green IoT. The system comprised three chaotic systems, generating a random data sequence half the size of the image to reduce energy consumption. Two distinct secret keys were utilized, and the sum of the plain images was calculated to generate a data sequence from the chaotic system. Encryption was carried out through two rounds of diffusion–permutation network operations, with the diffusion operation preceding the permutation. Each round employed a different secret key. It is worth noting that the time calculations were conducted on a normal PC rather than a constrained device to ensure accurate results.

Feng et al. [12] introduced a hybrid chaotic encryption with dynamic precision tailored for the IoT domain. To address the challenge of quantization precision in a single chaotic map system, they employed a random perturbation strategy between logistic and tent chaotic systems. A unified hardware architecture was designed to ensure the encryption algorithm's applicability under various precision conditions. The chaotic data sequence was converted to a bitstream and XORed with 8 bits of the image pixel. The study found that a reduction in precision led to a rapid decline in both sequence randomness and hardware overhead of the system. Recently, several proposed chaotic image encryption methods have been found insecure and susceptible to various cryptanalysis techniques [13], [14], [15].

To address the limitations in existing image encryption techniques within IIoT environments and overcome common weaknesses, such as the computational complexity of chaotic models, dependence on the plaintext for the secret key, and the cost of permutation operations during encryption, this study aims to introduce a novel image encryption algorithm. The primary objective of this work is to develop an encryption method that fulfills the security demands of IIoT data. The algorithm ensures secure image transmission with single-round encryption on the IoT camera sensor and the ability to encrypt multiple images using a single secret key in a short time frame. Cyclic construction is employed, utilizing the perturbed logistic chaotic map to maintain a comparable number of mathematical operations to the original logistic chaotic map, thereby enhancing the security of the data sequence and facilitating one-round cipher image generation.

Three matrices, matching the size of the plain image, are generated. Two of these matrices are dedicated to creating the permutation matrix. Obtained in a pre-encryption process, these matrices can be reused for multiple images captured by the IoT camera sensor. The permutation matrix is formed based on the direction and movement between chaotic state positions, mimicking an empty plain image. This novel permutation demonstrates high randomness, making it challenging to predict patterns and effectively break correlations between adjacent pixels. The permutation and diffusion processes work concurrently in the encryption, encapsulated within a single round.

Two diffusion functions are employed to ensure that both nonlinear and linear diffusion properties are achieved during the encryption process. Experimental results conducted on two devices, encryption on the loT camera sensor with limited memory and processing power, and decryption on a normal PC acting as an edge device, provide fair and insightful outcomes. The proposed algorithm demonstrates its capability to efficiently encrypt images of any size and type in a short time frame on both devices, generating secure images resistant to a set of attacks. The main contributions can be summarized as follows.

1) Introducing cyclic perturbed logistic map (CPLM), a new chaotic model, designed to generate a highly random chaotic data sequence while maintaining the computational efficiency of the original logistic chaotic map.
2) Proposing a new image encryption technique for secure IIoT data transmission to address industrial challenges. The algorithm incorporates diffusion and confusion characteristics within a single round, allowing a single secret key to encrypt multiple captured images in a single session.
3) Experimental results demonstrate that the proposed CPLM and the image cipher exhibit elevated levels of security and randomness compared with existing algorithms, effectively resisting numerous well-known attacks.

The rest of this article is organized as follows. Section II elaborates on the proposed secure IIoT data communication framework. Section III delves into the cycle chaotic map and presents the experimental results. Section IV introduces a new image encryption algorithm, while Section V discusses the experimental results and performance. Finally, Section VI concludes this article.

II. Securing IIoT Data Communication Framework

IoT camera sensors play a crucial role in monitoring and surveillance within IIoT environments, particularly in industrial zones [16]. They enable continuous image capture, aiding in the establishment of alarm systems, and are deployed in high-risk areas. However, due to power and memory constraints, ensuring

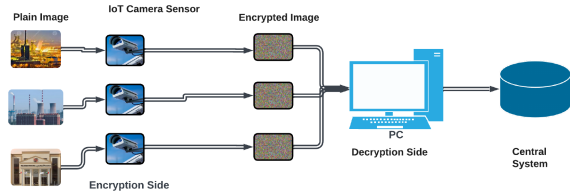


Fig. 1. Secured IIoT data communication framework.

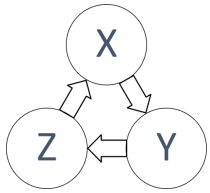


Fig. 2. Construction of CPLM with three nodes.

secure image sharing is challenging. As a result, IoT camera sensor-based surveillance is tailored for specific objectives, capturing high-resolution images with large dimensions [17]. To protect sensitive information, encryption is essential before transmission over wired or wireless communication technologies.

The secured IIoT data communication framework comprises the following three key components.

1) *IoT Camera Sensors:* These constrained devices are strategically distributed throughout industrial zones, capturing digital images and sharing them with other edge devices. They possess the capability to move around zones and utilize zoom functionalities.
2) *Edge Device:* Serving as a link between IoT camera sensors and the industrial network, the edge device is equipped with power and memory resources. It performs preanalysis, filtering, and classification of captured images and data from cameras before transmitting them to the central system. On this device, decryption and data processing can be implemented, reducing latency and bandwidth requirements.
3) *Industrial Network:* Representing the broader network infrastructure within the industrial environment, the industrial network connects IoT cameras, edge devices, and central systems, such as cloud computing or server-based systems. It facilitates data communication between cameras, edge devices, and the central system for subsequent storage, analysis, and processing.

Fig. 1 illustrates the secured IIoT data communication framework designed for an industrial zone. The framework comprises three essential components. An attacker may exploit insecure communication between IoT cameras and edge devices, executing various cyber attacks, such as modifications and insertions. Secure communication involves more than just encryption; it encompasses several key components, including authentication and authorization, integrity checking, key management, intrusion detection, and response, among others. This article concentrates on securing captured images transmitted between IoT cameras and edge devices, while additional security requirements can be addressed using lightweight cryptographic solutions due to the relatively small size of the data compared with digital images [5].

III. Cycle Chaotic Map

In this section, the perturbed logistic map and the new CPLM take center stage, playing major roles in the proposed encryption algorithm. The section provides an assessment of the performance of CPLM.

A. Perturbed Logistic Chaotic Map

The logistic chaotic map is susceptible to security vulnerabilities, making chaotic encryption vulnerable to attacks. Consequently, the recently enhanced logistic map offers improved security while maintaining the computational complexity of the original logistic map. The perturbed logistic map can be expressed mathematically as [7]

$$x_{i+1} = \left| r \times \left(1 - \frac{r}{x_i}\right) \right| \bmod 1 \tag{1}$$

where the x_i is a chaotic state and r is a control parameter. The perturbed logistic map exhibits full chaotic behaviors across the range $(0, \infty)$ [7].

B. Cyclic Perturbed Logistic Map

CPLM comprises three nodes, each hosting a version of the perturbed logistic map, as shown in Fig. 2. These maps, associated with different chaotic variables (initial condition and control parameter), collectively generate a highly perturbed map with robust chaotic security [7], [18]. Each node N denotes as $X(i, r_X)$, $Y(i, r_Y)$, and $Z(i, r_Z)$. The chaotic state of each node contributes to the generation of the next chaotic state in the subsequent node. The transition between nodes follows a cyclic pattern in a clockwise direction. To enhance the unpredictability of the generated sequence, a perturbation function is employed before utilizing the current chaotic state within the same node. This perturbation function is mathematically expressed as

$$N_{X(i,r_X)} = N_{X(i-3,r_X)} + N_{Z(i-1,r_Z)} \bmod 1 \tag{2}$$

$$N_{Y(i,r_Y)} = N_{Y(i-3,r_Y)} + N_{X(i-1,r_X)} \bmod 1 \tag{3}$$

$$N_{Z(i,r_Z)} = N_{Z(i-3,r_Z)} + N_{Y(i-1,r_Y)} \bmod 1. \tag{4}$$

The notation N_i designates the current node, with $i-1$ referring to the preceding node. $N_{X(i-3,r_X)}$, $N_{Y(i-3,r_Y)}$, and $N_{Z(i-3,r_Z)}$ represent the states generated two iterations prior in the current node, while $i-1$ denotes the chaotic state generated by the node just before. Subsequently, $N_{X(i,r_X)}$, $N_{Y(i,r_Y)}$, and $N_{Z(i,r_Z)}$ represent the newly generated chaotic state of the current node, acting as input to the perturbed logistic map.

In each node, there is a buffer to store the generated chaotic state. When the CPLM visits the node, the stored chaotic state in the buffer is used in the perturbation function. When CPLM visits $N_{X(i,r_X)}$, the $X(i-3, r_X)$ state was stored in the buffer

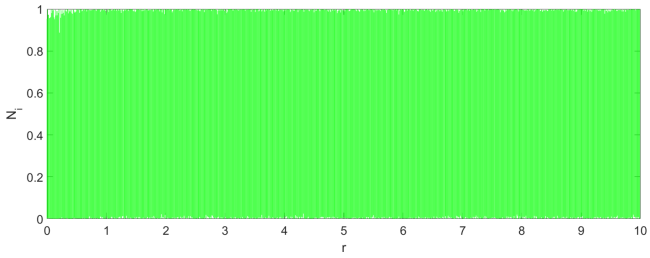


Fig. 3. Bifurcation diagram of CLPM.

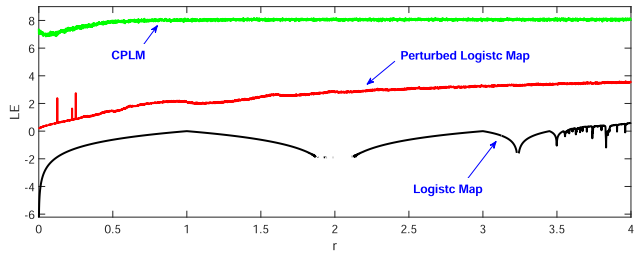


Fig. 4. LE values of the three chaotic maps.

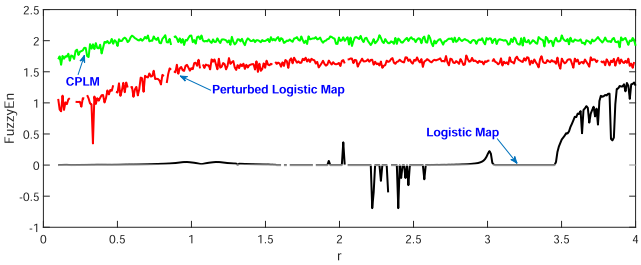


Fig. 5. Fuzzy entropy values of the three chaotic maps.

to be used in node X during the current iteration i. The CPLM commences from node X, moves to Y and Z, and then returns to X in a cyclical transition. When visiting each node, one chaotic state (chaotic point) is generated, resulting in a 1-D data sequence of chaotic points.

To assess the chaotic behavior of the CPLM, various metrics including bifurcation, Lyapunov exponent (LE), and fuzzy entropy are employed. Fig. 3 illustrates the fully shaded region across parameter settings ranging from 0 to 10. This emphasizes the absence of nonchaotic behavior or fixed points, indicating a lack of predictability, as random numbers are challenging to anticipate.

In the context of the LE, high positive values signify a heightened sensitivity to initial conditions within chaotic states, resulting in significant divergence over time. This characteristic defines chaotic behavior and randomness, with LE serving as a numerical identifier of chaotic dynamics. Fig. 4 further highlights the distinctions among three chaotic maps: logistic, perturbed logistic, and CPLM, within the range of 0–4, mirroring the original range of the logistic map. CPLM exhibits elevated values indicative of robust randomness.

Fuzzy entropy, serving as a metric for complexity, can detect the embedding patterns between chaotic states in a data sequence. When there are no embedding patterns present, a larger numerical positive value is demonstrated. In Fig. 5, CPLM demonstrates superior fuzzy entropy, making it challenging to discern patterns in the data sequence generated by CPLM. Consequently, CPLM exhibits enhanced security, approximating the number of calculations required in comparison with the original logistic map.

TABLE I
PERIOD LENGTH ACROSS CHAOTIC VARIABLES (CONTROL PARAMETERS $r_X = 3.9$, $r_Y = 4.9$, AND $r_Z = 5.9$), INITIAL CONDITION (0.8) FOR THREE NODES

Bit Precision	Period Length	LE	Fuzzy Entropy
8 bits	499	0.87	0.42
10 bits	76 867	0.98	0.54
12 bits	25 571	1.14	0.74
14 bits	$>2^{25}$	3.45	1.1
16 bits	$>2^{25}$	4.75	1.24
32 bits	$>2^{25}$	5.31	1.87

Reducing data precision in digital chaos leads to dynamical degradation, especially evident in IIoT camera sensors with low-bit precision, such as 16 bits. Chaotic maps degrade with reduced precision, causing nonrandomness, lack of distribution, and increased correlation in short data sequences [19]. Longer periods in chaotic data sequences indicate less degradation. Table I outlines period length across chaotic variables across various bit precisions. Increasing the bit precision enhances CPLM's period length due to the three nodes and the perturbation function in (2)–(4), which generates jumps between chaotic points in a single data sequence, thereby enhancing randomness.

IV. IIoT CHAOTIC IMAGE ENCRYPTION

Images in the IIoT are frequently captured by imaging sensors within the industry's smart space and are directly transmitted to edge devices. It is crucial to ensure the security of these images during transmission. Given that image sensors are often constrained devices with limitations in memory and power, and considering the challenges posed by a poor communication environment, the encryption algorithm must be both highly efficient and secure. To address these constraints, the algorithm should prioritize speed and security simultaneously. This necessitates a reduction in the number of rounds, and calculations, and the avoidance of unnecessary operations to accommodate the resource limitations of image sensors, ensuring optimal performance in real-time scenarios.

Image encryption involves two distinct phases: pre-encryption and the encryption process itself. In the pre-encryption phase, a key is utilized to generate a chaotic data sequence, which is then employed in subsequent operations. In addition, a permutation matrix and a diffusion matrix are generated based on the size of each image. These matrices play crucial roles in the subsequent encryption process. In the second phase of encryption, simultaneous diffusion and permutation operations are applied to enhance the security of the image, ensuring robust protection of the image data in a short time.

A. Re-Encryption Process

In this section, two operations are implemented: one for generating chaotic data sequences and another for creating permutation and diffusion matrices.

1) Data Sequence Generation: CPLM comprises three nodes, each equipped with a perturbed logistic map featuring three distinct chaotic variables. Six variables can be randomly selected, with the control parameter ranging from 0 to 10 and initial conditions spanning from 0 to 1. The secret key, crucial for robust security, can be any bitstream with a length greater than 128 bits to resist brute force attacks. In the encryption process, the secret key is employed to perturb the three initial conditions using these formulas [7]

$$\begin{aligned} N_{X(i,r_X)} &= \frac{N_{X(i,r_X)} + \frac{1}{\sum_{L=1}^{i}(K_L)}}{2} \\ N_{Y(i,r_Y)} &= \frac{N_{Y(i,r_Y)} + \frac{1}{\sum_{L=1}^{i}(K_L)}}{2} \\ N_{Z(i,r_Z)} &= \frac{N_{Z(i,r_Z)} + \frac{1}{\sum_{L=1}^{i}(K_L)}}{2} \end{aligned} \tag{5}$$

where (K_L) represents the bitstream in the secret key. In each iteration i, the chaotic state is modified to obtain a new value based on the summation of the secret key $\frac{1}{\sum_{L=1}^{i}(K_L)}$. The chaotic state can take any value when $i = 1$. Equation (5) can be implemented based on the size of the secret key. For instance, assuming that the secret key is 156 bits, (5) is executed 156 times. Subsequently, three fully new chaotic states are generated in the three nodes, each possessing a 52-bit significant size based on fixed-point representation. These chaotic states can also be utilized on the fractional side for controlling parameters to increase sensitivity to the secret key. Although the secret key can be initialized with any size between 156 and 312 bits, having a key length of 312 bits allows for the generation of six chaotic variables (three for initial conditions and three for control parameters).

The CPLM can generate a data sequence by using three initial conditions and three control parameters. This data sequence consists of chaotic states with sizing corresponding to those of a plain image. The CPLM is implemented with a size of $M \times N \times W$, where M and N represent the height and width of a plain image, and W can be either one channel or three channels, depending on the image type (gray or RGB). The chaotic data sequence is employed to create three distinct data sequences: two for permutation and one for diffusion. Assuming the generated original data sequence D_i, three data sequences can be generated from the original data sequence using the following expressions [20]:

$$A_i = D_i \times 2^{13} \mod 8 \tag{6}$$

$$B_i = D_i \times 2^{14} \mod M + 1 \tag{7}$$

$$C_i = D_i \times 2^{15} \mod 256 \tag{8}$$

where A_i and B_i are used for permutation operations to generate a new permutation matrix with the same size as the plain image. C_i is a diffusion matrix with integer chaotic states ranging between 0 and 255 to use in diffusion operation.

2) Permutation Matrix: In this operation, a permutation matrix is generated using the random indices of the plain image. Assuming two empty sample images of the same size as the plain image, each pixel in the sample image is assigned two addresses: one related to height and another related to width. Here, A_i is used to determine the direction, as shown in Fig. 6, and B_i is used to count how many movements are made from the current pixel in the same direction as A_i. The permutation matrix stores the indices of the height and width of each visited pixel in the sample image.

As each pixel in the sample image is visited, a value of -1 is assigned to mark the pixel as visited and prevent revisiting. If a pixel with a value of -1 is encountered again, the movement is skipped, and the next value is selected from the two matrices, A_i and B_i. Consequently, some pixel values may remain unvisited and will be left empty without a -1 value. These unvisited values are then visited row by row, starting from the first row to the end row, and their indices are recorded in the permutation matrix.

In the permutation matrix, the movement is conducted row by row, while in the sample image, the movement is determined randomly based on the values from A_i and B_i. The initial pixel value, representing the starting point for movement, can be any pixel; in this article, the midpoint is used. Finally, the permutation matrix P_M has new indices of the pixels. In each cell, two indices values are stored. In the following operations, the permutation operation can be applied directly based on the permutation matrix. The plain image is then reread based on the values in the permutation matrix.

B. Encryption Process

The preceding steps constitute the preprocessing phase, and IIoT devices can generate and reuse these operations multiple times in the encryption algorithm. The simultaneous diffusion and permutation operations are outlined as follows.

1) The first eight pixels of P_D are taken from the first row of the plain image P, and the last eight pixels of P_D are taken from the last row of P. These sixteen pixels are XORed directly with their corresponding values in the C_i matrix. P_D has eight values in the first row and eight values in the last row.
2) One pixel is taken from each side of P, copied to P_D, XORed with the corresponding value of C_i, and then the encryption function is applied to achieve linear and nonlinear diffusion. The diffusion operations are governed by the following equations [10]:

$$P_{Di} = P_{Di} \oplus C_i \tag{9}$$

$$P_{Di} = \left(\sum_{n=i-8}^{i} ((P_{Di})_n + C_n) \right) \mod 256. \tag{10}$$

3) Pixels are retrieved row by row from both sides until reaching the middle value in P_D. This process creates two diffusion lines, one starting from the beginning and

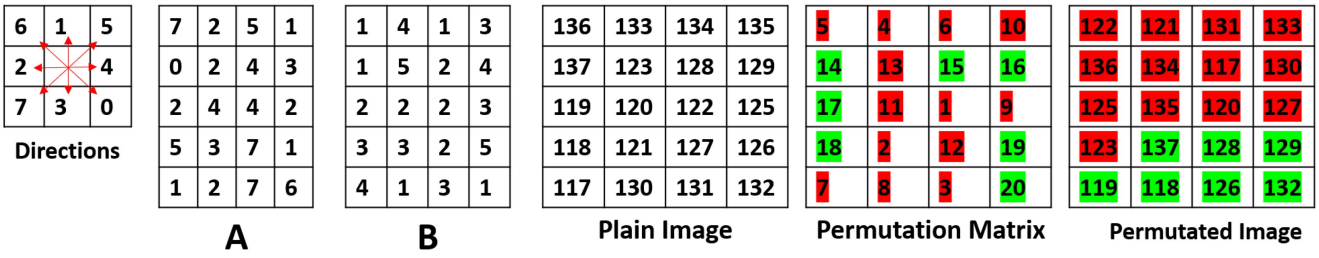


Fig. 6. Permutation matrix and example.

another starting from the end. Both lines converge at the middle point.

4) Permutation operations are initiated when the two lines converge at the middle value in P_D, as all pixels have already been read based on row-by-row in P before.
5) A new pixel after the middle value in P_D in the two lines is encrypted again based on (9) and (10) and moved to matrix P_C according to corresponding values in the permutation matrix P_M. Simultaneously, the diffusion–permutation processes start from the middle point on two sides.
6) Only a single round involving permutation and diffusion operations is utilized. The diffusion operations have two sides, one commencing from the beginning and the other from the end.

One round comprises two operations, as there are two sides. The permutation operation relies on the random CPLM, breaking the correlation between pixels. The permutation matrix can be generated once and used for multiple images, requiring minimal computations during the encryption process. It involves reading the addresses inside each cell and transferring the corresponding pixels from P_D to P_C. On the other hand, the diffusion operations depend on two functions, XOR and MOD, providing linear and nonlinear diffusion on the two sides. Each pixel in each plain image encrypts twice, influencing subsequent pixels 16 times (eight from top to bottom and eight from bottom to top). Diffusion and confusion are applied simultaneously in one round, ultimately generating the fully encrypted image.

Algorithm 1: Image Encryption.

Data: Image, P and Secret key, K
Result: Encrypted Image (P_C)

$[COL, ROW, CA] = size(P, 1, 2, 3)$;
for $i = 1$ *to* $ROW \times COL \times CA$ **do**
  $D_i = CPLM(K, N_X, N_Y, N_Z)$;
  $A_i = D_i \times 2^{13} \ mod \ 8$;
  $B_i = D_i \times 2^{14} \ mod \ ROW + 1$;
  $C_i = D_i \times 2^{15} \ mod \ 256$;
$P_M = Permutate\ (A, B)$;
for *Channel=1 to CA* **do**
  $P_D = P(:, :, Channel)$;
  for *i=1 to COL* **do**
    for *J=1 to ROW* **do**
      $P_D(i,j) = P_D(i,j) \oplus C(i,j)$;
      $P_D(i,j) = \sum_{n=i-8}^{i}(P_D(i,j)_n + C(i,j)_n) \ \mathrm{mod}\ 256$;
      $P_D(COL-i, ROW-j) = P_D(COL-i, ROW-j) \oplus C(COL-i, ROW-j)$;
      $P_D(COL-i, ROW-j) = \sum_{n=i-8}^{i}(P_D(COL-i, ROW-j)_n + C(COL-i, ROW-j)_n) \ \mathrm{mod}\ 256$;
      if $i >= mid_i \ \&\ j >= mid_j$ **then**
        $P_C(i,j) = P_D(P_M(i,j))$;
        $P_C(COL-i, ROW-j) = P_D(P_M(COL-i, ROW-j))$;
  $P_C(:, :, Channel) = P_C$;

C. Decryption Process

In the decryption phase, the device is not constrained, whether it is a cloud computing system, mobile application, or server. These devices have ample power and memory resources compared with sensor devices or surveillance devices. Decryption is faster due to the abundance of resources. The decryption algorithm consists of two phases: predecryption and decryption. In the predecryption process, the secret key is utilized to generate three matrices in the same manner as in the pre-encryption process. During the decryption process, the cipher image is reread based on the inverse permutation matrix IP_M. To generate IP_M, P_M is a matrix with row and column actual indexes, where each index corresponds to a cell. Each cell contains two indexes (row and column). The swapping between stored indexes in P_M with actual indexes generates IP_M. The diffusion operations begin from both sides (bottom to top and top to bottom). The decryption function can be expressed as follows [10]:

$$P_{Di} = \left(\sum_{n=i-8}^{i-1} ((P_{Di})_n + C_n) + C_i - P_{Di} \right) \mod 256 \tag{11}$$

$$P_{Di} = P_{Di} \oplus C_i. \tag{12}$$

Decryption continues until both the beginning and the end of the cipher image are reached. This completes the decryption process, and the cipher image is entirely decrypted to generate a recovered image. Algorithms 1 and 2 provide a clear and understandable representation of the encryption and decryption processes, respectively.

Algorithm 2: Image Decryption.

Data: Encrypted Image (P_C) and Secret key, K
Result: Image, P

$[COL, ROW, CA] = size(P_C, 1, 2, 3)$;
for $i = 1$ *to* $ROW \times COL \times CA$ **do**
  $D_i = CPLM(K, N_X, N_Y, N_Z)$;
  $A_i = D_i \times 2^{13}$ *mod* 8;
  $B_i = D_i \times 2^{14}$ *mod* $ROW + 1$;
  $C_i = D_i \times 2^{15}$ *mod* 256;
$P_M = Permutate\ (A, B)$;
for *Channel=CA to 1* **do**
  for *i=COL to 1* **do**
    for *J=ROW to 1* **do**
      $P_D(i,j) = P_C(IP_M(i,j), Channel)$;
  for *i=COL to 1* **do**
    for *J=ROW to 1* **do**
      $P_D(i,j) = \sum_{n=i-8}^{i-1}(P_D(i,j)_n + C(i,j)_n) + C(i,j) - P_D(i,j) \mod 256$;
      $P_D(i,j) = P_D(i,j) \oplus C(i,j)$;
      $P_D(COL-i, ROW-j) = \sum_{n=i-8}^{i-1}(P_D(COL-i, ROW-j)_n + C(COL-i, ROW-j)_n) + C(COL-i, ROW-j) - P_D(COL-i, ROW-j) \mod 256$;
      $P_D(COL-i, ROW-j) = P_D(COL-i, ROW-j) \oplus C(COL-i, ROW-j)$;
  $P(:,:,Channel) = P_D$;

Fig. 7. IoT camera sensor device.

V. Experimental Results and Performance

In this section, all experiments were conducted on an IoT camera device equipped with a quad-core 32-bit A7 CPU, a high-performance quad-core AI vision processor, a USB Camera, 1.5 GHz, and 4 GB RAM, as shown in Fig. 7. The portable nature of the device allows for the use of any operating system and supports the Python language. The experimental tests were scripted in Python and executed on the IoT camera device. The resulting image used in the experiments has dimensions of 3264 pixels (height) by approximately 2448 pixels (width), representing a large digital image size with high-quality full HD resolution. Two images were captured, and the Pepper image was selected due to its common usage in state-of-the-art image encryption algorithms. To ensure a fair comparison with other algorithms and to study the encryption algorithm thoroughly, commonly used images in the domain were employed.

Fig. 8 demonstrates the efficacy of the proposed algorithm in producing cipher images that resemble noise. The encryption process has been carefully crafted to incorporate both confusion and diffusion within a single round. Modifying a single bit of the secret key results in a completely new cipher image with a substantial dissimilarity. In the case of color images, the three channels (red, green, and blue) are individually encrypted. Subsequently, the proposed algorithm is applied, and the cipher image is formed by amalgamating all three channels. The resultant colored image pixels are independently shuffled within each channel, effectively eliminating any statistical remnants of the original image.

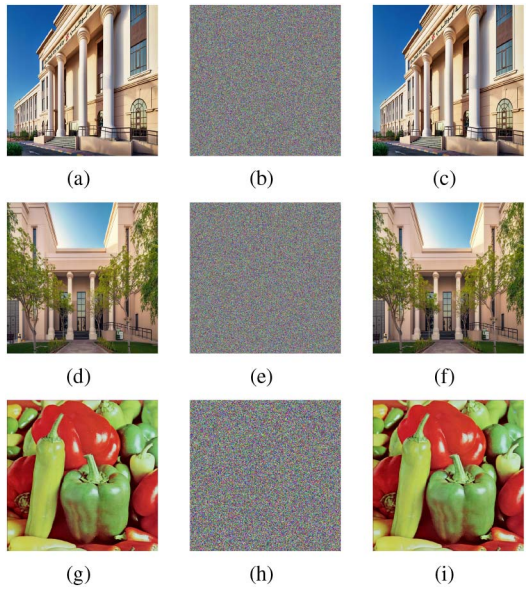


Fig. 8. Three plain images with corresponding cipher images: (a) Campus 1 size $3264 \times 2448 \times 3$, (b) encrypted image, and (c) decrypted image, (d) Campus 2 image size $3264 \times 2448 \times 3$, (c) encrypted image, and (d) decrypted image, and (g) Pepper image size $512 \times 512 \times 3$, (h) encrypted image, and (i) decrypted image.

A. Histogram Analysis

A histogram is a statistical test used to quantify the frequency of pixel values in a cipher image. Given that pixel values typically range from 0 to 255, the count of occurrences provides insights into the image, particularly when comparing known images or images captured under similar conditions. The objective of a histogram is to ensure that the occurrences of pixel values are distributed evenly, aiming to eliminate any discernible patterns or data leakage from the cipher image. The ideal scenario is a flat histogram. In Fig. 9, the histograms of two distinct plain images and their corresponding cipher images are presented. It is evident that the histograms of the cipher images closely approximate a flat distribution. This underscores the capability of the proposed algorithm to generate secure images without revealing patterns or leaking information in the ciphered results.

B. Correlation Coefficients (CC)

The CC is a statistical measure focused on studying the relationship between adjacent pixels in an image. In the plain image, the relationship is very strong, indicated by a low percentage of change in pixel values, and many regions in the image have the

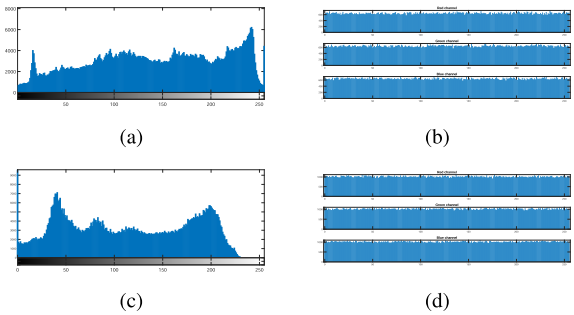


Fig. 9. Histogram of (a) Campus 1 (plain image), (b) Campus 1 (cipherimage), (c) Pepper (plain image), and (d) Pepper (cipherimage).

TABLE II
CC FOR VARIOUS IMAGES

Image	plain image			Cipherimage		
	D	V	H	D	V	H
Campus 1	0.9874	0.9351	0.8475	−0.00023	−0.00012	−0.0003
Campus 2	0.9124	0.8698	0.8547	−0.00016	−0.0021	−0.0012
Pepper	0.9764	0.9524	0.9582	0.00021	0.00012	0.0002

TABLE III
CC COMPARISON OF THE ENCRYPTED IMAGE

Image	Cipherimage		
	D	V	H
Proposed algorithm	**0.00021**	**0.00012**	**0.0002**
Ref. [7]	0.0003	0.0006	0.005
Ref. [21]	0.0011	0.0012	0.016
Ref. [22]	0.0004	0.0011	0.0125
Ref. [19]	0.0015	0.0013	0.012

same value. The CC expresses these relationships as a numerical value, where a CC value near 0 signifies no correlation, and values near 1 or −1 indicate high correlation. The computation of the CC for four images is presented in Table II, along with results from other image encryption algorithms in Table III. The proposed algorithm exhibits low values close to 0, indicating minimal correlation between pixels in the cipher image. This low CC value can be attributed to two primary factors. First, the permutation operation, which is based on the CPLM and the specific method employed for generating the permutation matrix. Second, the diffusion functions involving XORing and modular operations are applied to nine pixels with C values and influenced by the values of the diffusion matrix. This process contributes to the suppression of correlation between pixel values.

C. Chosen/Known Plaintext Attacks

This measurement method belongs to cryptanalysis and is employed to assess an algorithm's vulnerability. In this approach, attackers aim to recover the secret key by discerning hidden recurring patterns through the comparison of plain and cipher images. To test the resilience of the proposed algorithm, black and white images are commonly utilized to identify potential weaknesses in the resulting cipher image. If the encryption algorithm successfully generates a cipher image with random

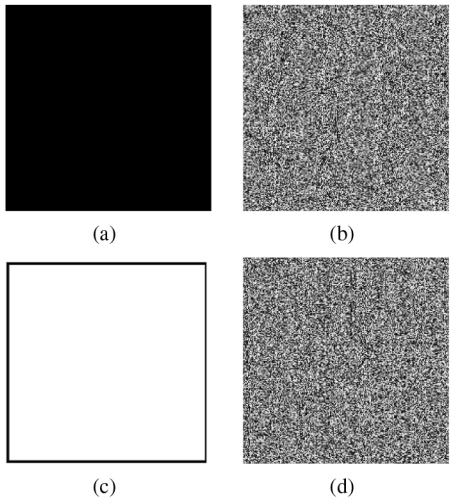


Fig. 10. Encryption results of black and white images: (a) black plain image, (b) black cipher image, (c) white plain image, and (d) white cipherimage.

values derived from black and white plain images, it signifies a robust encryption process capable of withstanding this form of attack. Conversely, if cipher images exhibit noticeable patterns, they may be susceptible to exploitation by attackers, allowing them to deduce a pathway for key extraction. In such cases, these types of encrypted images demonstrate vulnerabilities to this specific type of cyber attack.

Fig. 10 displays two plain images alongside their corresponding cipher images. The two cipher images resemble full noise-like and highly random images. This result can be attributed to the significant role of the diffusion matrix in generating randomness. First, the diffusion matrix comprises random values generated by CPLM, which are then utilized in the modular function with the preceding eight pixels. In addition, encrypting the image from both sides contributes to the creation of high random values.

D. Differential Attack Analysis

In evaluating the sensitivity of plain images to minute changes in pixel values, the differential attack is a prevalent method within this context. A robust encryption algorithm should exhibit resistance to this attack, ensuring that a minor alteration in the plain image results in a distribution across the entire cipher image, rather than concentrating in a specific region. To achieve this, two diffusion functions are employed on both sides, ensuring that the cipher image is intricately linked and any change to one pixel propagates throughout the entire image. This prevents attackers from identifying recurring patterns or region-specific patterns across different versions of plain images.

When evaluating susceptibility to a differential attack, two key metrics are often utilized: the number of pixel change rate (NPCR) and the unified average change intensity (UACI). These metrics involve modifying a single bit in the plaintext image while maintaining uniform usage of the secret key for both images during encryption. NPCR and UACI can be estimated

TABLE IV
NPCR AND UACI VALUES

	Image	NPCR	UACI
Proposed algorithm	Campus 1	99.81	33.46
Proposed algorithm	Campus 2	99.66	33.50
Proposed algorithm	Pepper	99.68	33.48
Ref. [23]	Various images	99.60	33.46
Ref. [24]	Various images	99.60	33.42

as

$$\text{NPCR}(C_1, C_2) = \frac{\sum_{i,j}^{M,N} D(i,j)}{MN} \times 100\% \tag{13}$$

and

$$\text{UACI}(C_1, C_2) = \frac{1}{MN} \sum_{i,j}^{M,N} \frac{|C_1(i,j) - C_2(i,j)|}{L} \times 100\% \tag{14}$$

respectively, where MN is the total number of pixels in a plain image, $L = 255$ is the maximum gray level value for an 8-bit pixel, and

$$D(i,j) = \begin{cases} 1, & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0, & \text{if } C_1(i,j) = C_2(i,j). \end{cases} \tag{15}$$

Table IV presents the results of NPCR and UACI values for three images in comparison with other existing algorithms. The high values observed for both metrics underscore the resistance of the cipher image to the differential attack. The encryption process ensures that a change in a single pixel propagates across the entire image, making it challenging for an attacker to discern any discernible patterns. This is achieved by interlinking nine-pixel values together on both sides, with each side influencing the other, creating a cohesive and interconnected image without distinct regions.

E. Speed Analysis

Speed is a crucial metric for algorithms, especially for constrained devices with limited power and memory. To assess the speed of the proposed algorithm, two devices with different speeds were used in the experiments. As mentioned earlier, the IoT camera device was used for encryption, as it captures images and sends them to edge devices, normal PCs, or mobile devices. In the experiments, decryption was performed on a PC with an Intel Core i7 CPU @ 2.80 GHz, 2.90 GHz, and 16.0 GB RAM. By default, decryption is faster than encryption due to differences in power and memory. The pre-encryption and predecryption phases are not included in this measurement, as they generate matrices before the encryption process and can be used to encrypt multiple images before the user decides to change the secret key for the session. Therefore, the focus is on the speed of the encryption process to showcase the efficiency of the proposed algorithm in constrained devices.

Table V illustrates the speed of the encryption and decryption algorithms for two images of varying sizes. The encryption process takes more than 1 s, while the decryption process takes less than 1 s. This efficiency is attributed to the simplicity of the encryption and decryption algorithms, which involve only three operations: two functions for diffusion and one for moving pixels from one position to another. The XOR operation and modulo operations require a low number of calculations, enabling swift execution.

TABLE V
SPEED ANALYSIS OF ENCRYPTION AND DECRYPTION ON AN IOT CAMERA DEVICE AND NORMAL PC

Image	Encryption (s)	Decryption (s)
Campus 1 ($3264 \times 2448 \times 3$)	12.35	0.817
Pepper ($512 \times 512 \times 3$)	0.403	0.065

To estimate the theoretical time for encrypting an image on an IoT image sensor, consider the following specifications: Processor clock rate: 1.5 GHz (1.5 billion cycles per second). For each diffusion operation, the summation requires approximately 306 cycles, with 18 numbers and 17 cycles for each summation. The modular operation requires five cycles, and the XOR operation requires one cycle. Thus, the total number of cycles is 306. Therefore, the theoretical execution time for one diffusion process is approximately 0.000000208 s. For the Pepper image, the approximate time is 0.327155712 s, which is close to the real-time value in Table V.

F. Key Sensitivity and Space

The key space of the proposed algorithm is 156 bits [25], which remains secure and resistant to brute force attacks. However, it is important to study the sensitivity of the key to changes to assess the algorithm's confusion property. To conduct this test, a plain image was encrypted twice using two different keys: key k_1 had "01" repeated 78 times to generate a 156-bit key, whereas key k_2 had "01" repeated 78 times with the last bit flipped to "0." Two ciphertexts, C_1 and C_2, were generated using these keys. The results are presented in Fig. 11. The proposed algorithm demonstrates a high sensitivity to changes in just one bit of the key, leading to the generation of a new cipher image. This sensitivity arises from the chaotic nature of the CPLM. Alterations in the initial conditions of the CPLM lead to markedly different data sequences, consequently impacting the resulting encrypted image.

The secret key is initially 156 bits long but can be extended to a duplicate length of 312 bits, enabling it to serve as control parameters and initial conditions. This extension amplifies the sensitivity of the secret key for the proposed image encryption. Experimental findings demonstrate that both 312-bit and 156-bit key lengths show similar sensitivity. Hence, opting for a 156-bit key length is adequate, as it yields comparable results to duplicating the key length to 312 bits.

G. Comparison

In the comparison section, a set of images was selected to align with other researchers in the field, incorporating measurements of speed, CC, NPCR, and UACI in the comparison.

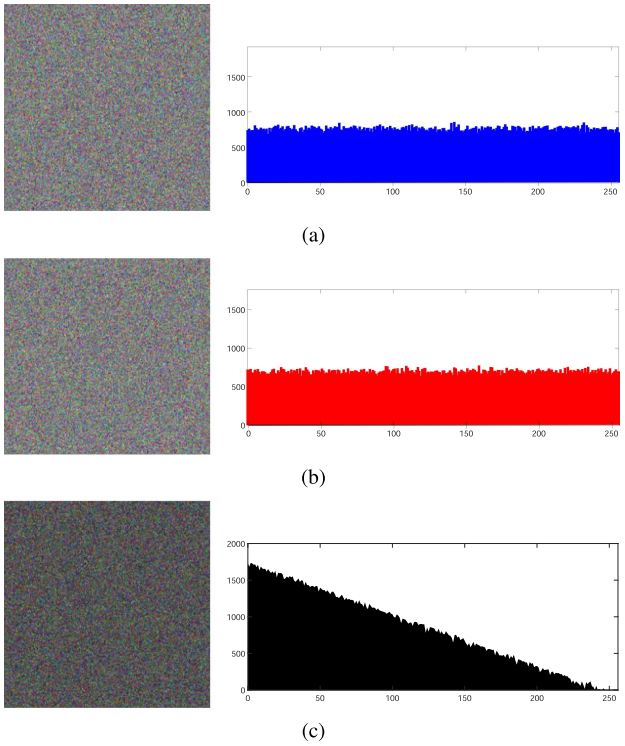


Fig. 11. Key sensitive during encryption. (a) Cipher image C1 using K1. (b) Cipher image C2 using K2. (c) Difference between cipher C1 and C2.

To demonstrate the superiority, performance, and efficiency of the proposed algorithm, a selection of recent research featuring various image encryption algorithms was chosen. The other algorithms were reimplemented on the same devices to ensure a fair comparison and to evaluate performance on constrained devices. The experiments involved encrypting plain images of various sizes and types, facilitated by the SC-SIPI "Miscellaneous" image dataset [26], which consists of 28 sample images. These images vary in size, with dimensions, such as 256×256 and 1024×1024.

The algorithms proposed by Alawida [7], Azam et al. [23], and Zhou et al. [24] were primarily designed for conventional images and standard computers. In contrast, the image encryption algorithm proposed by Feng et al. [12] is tailored for IoT devices with varying bit precisions. Azam et al. [23] focused on generating S-boxes using pseudorandom number generation in their image encryption scheme, while Alawida [7] and Zhou et al. [24] utilized chaotic maps with novel permutations and diffusion algorithms in their designs.

The proposed algorithms offer advantages, such as encryption in a single round, and demonstrate better security merits. Diffusion is achieved bidirectionally, and the permutation relies on the security of the CPLM. Table VI summarizes the comparison results, highlighting that the proposed algorithm excels in both security and speed compared with others. Table VII summarizes the average performance comparison between the proposed algorithm and existing methods reported in the literature. The proposed algorithm achieves competitive results on par with recent algorithms.

TABLE VI
COMPARISON RESULTS ON THE SAME MACHINES

Algo	Image	Test		
		CC	NPCR/ UACI	Speed Second (E/D)
This article	**Campus 1**	-0.00021	99.812/ 33.465	12.35/ 0.817
	Campus 2	0.0002	99.664/ 33.504	12.35/ 0.817
	Pepper	0.0002	99.682/ 33.483	0.403/ 0.065
	28 images	0.0013	99.662/ 33.475	13.681/ 0.912
Ref. [23]	**Campus 1**	0.0021	99.784/ 33.421	23.12/ 0.9842
	Campus 2	0.0013	99.874/ 33.687	23.14/ 0.9784
	Pepper	0.0014	99.622/ 33.487	5.214/ 1.124
	28 images	0.0014	99.689/ 33.435	122.12/ 26.15
Ref. [24]	**Campus 1**	0.0041	99.412/ 31.431	22.155/ 2.124
	Campus 2	0.0002	99.561/ 33.384	22.178/ 2.421
	Pepper	0.0002	99.698/ 33.451	1.9865/ 0.3254
	28 images	0.0014	96.412/ 33.145	29.124/ 8.1245
Ref. [7]	**Campus 1**	0.0004	99.678/ 33.632	19.241/ 1.865
	Campus 2	0.0001	99.487/ 33.978	19.121/ 1.897
	Pepper	0.0003	99.677/ 33.752	1.915/ 0.1994
	28 images	0.0004	99.612/ 33.453	28.142/ 7.121
Ref. [12]	**Campus 1**	0.0021	99.597/ 33.58	13.358/ 1.754
	Campus 2	0.0041	99.621/ 33.771	22.124/ 2.312
	Pepper	0.015	99.684/ 33.324	2.018/ 0.7604
	28 images	0.00152	99.654 33.358	31.754/ 10.231

TABLE VII
PERFORMANCE COMPARISON OF VARIOUS IMAGE ENCRYPTION CIPHERS

Algorithms	Comparison measurements			
	CC	NPCR	UACI	Keyspace
This article	0.0002	99.6671	33.4592	2^{156}
Ref. [27]	–0.00021	99.6084	33.4714	2^{298}
Ref. [28]	–0.00162	99.5937	33.4086	2^{512}
Ref. [29]	–0.0001	99.6102	33.4465	2^{256}
Ref. [30]	0.0009	99.6095	33.4640	2^{256}

H. Discussion

The proposed algorithm is designed to address the primary security vulnerabilities in IIoT devices that require secure and high-speed image encryption. Moreover, to overcome the limitations inherent in state-of-the-art image cipher algorithms for IIoT environments. It employs the CPLM as a source of entropy in a cyclic construction to generate random numbers and highly distributed values. In the cyclic construction, the same map is used for different chaotic states, and the perturbed functions are simple, adding minimal implementation burden.

The permutation matrix contributes to the proposed algorithm's full security, making it difficult for adversaries to determine the original positions of pixels. Each encrypted pixel after the middle point is moved to matrix P_C based on matrix P_M. As a result, encrypted pixels are distributed randomly in P_C, with no correlation between their positions. In addition, eight neighboring pixels are involved in encrypting one pixel after the middle point. These eight neighboring pixels are also randomly distributed in matrix P_C, making it difficult to decrypt the pixel without knowing matrix P_M. This is the first security

layer in the proposed algorithm that makes cryptanalysis very costly.

On the other hand, considering cryptanalysis of the diffusion operations alone, the proposed algorithm utilizes three simple operations—XORing, summation, and modular arithmetic—to achieve the diffusion property. Here, the adversary aims to reconstruct the original pixel values by summing the eight previous encrypted pixels with nine values from matrix C. Even if the adversary could guess all nine values of C and the eight previous encrypted pixels, the total number of possible combinations to generate these values is computationally infeasible to break. This is because each value has 2^8 (256) possibilities due to the 8-bit size, and there are 17 total values to guess (nine from C and eight previous pixels). Therefore, the total number of possible combinations is a massive $2(8 \times 17)$. This extensive calculation for each pixel, along with the multitude of possible combinations for the nine values of C, makes it practically impossible for the adversary to determine the correct combination without knowledge of the secret key, regardless of the permutation operation.

The proposed algorithm excels at statistical security, particularly against known plaintext attacks, and has fast encryption speeds. It efficiently encrypts images of any size and type, encrypting multiple images with a single secret key over extended sessions. This is especially useful for IIoT camera devices that communicate over wired and wireless channels, ensuring fast and secure data transmission in high quality. The algorithm's efficiency ensures secure image transmission, which aids in efficient communication and high-quality data delivery. A summary of the algorithm's important properties is as follows.

1) The proposed algorithm ensures key independence from the plaintext and can be utilized for multiple communications.
2) A single-bit change in the plain image causes a ripple effect throughout the entire cipher image, resulting in entirely distinct cipher images.
3) Modifying just one bit of the secret key results in a completely different cipher image.
4) Permutation and diffusion operations are tightly intertwined in a single round, influencing each other significantly.
5) The algorithm is versatile enough to handle various types and dimensions of images, as demonstrated in the comparison results.
6) The encryption provided by the proposed algorithm exhibits resilience against multiple attacks, including statistical, differential, and chosen/known-plaintext attacks.

VI. Conclusion

In this study, a novel chaotic image cipher was proposed to address vulnerabilities encountered by IIoT camera devices during image transmission to edge devices in industrial settings. The existing works have limitations in the chaos model with multiple calculations for constrained devices. In addition, there is not an independent key on the plaintext, and the permutation cost is high. In this article, these limitations are taken into consideration and addressed. The simple CPLM chaos model was utilized to generate three matrices, two of which were used to create permutation and diffusion matrices. Encryption was conducted in a single round, achieving simultaneous diffusion and confusion. Experimental results on two devices demonstrated enhanced security against various cyberattacks, such as histogram and differential analysis. Notably, the algorithm's efficiency on constrained devices enabled the encryption of multiple images with minimal delays in both wired and wireless communication. The proposed algorithm utilizes an independent key for multiple images, and the permutation matrix is constructed during preprocessing and utilized for multiple images. However, a limitation was observed in the algorithm's inability to be applied in parallel mode to enhance time efficiency. Therefore, future research should focus on exploring the challenge of implementing parallel encryption in a single round, particularly in IIoT environments.

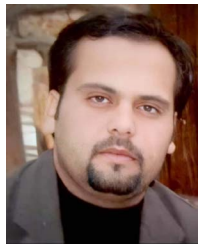
Acknowledgment

The author would like to thank all anonymous reviewers and editors for their invaluable contributions during the peer review process.

References

[1] M. A. Caraveo-Cacep, R. Vázquez-Medina, and A. H. Zavala, "A review on security implementations in soft-processors for IoT applications," *Comput. Secur.*, vol. 139, 2023, Art. no. 103677.
[2] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: A systematic literature review," *IEEE Access*, vol. 8, pp. 120331–120350, 2020.
[3] P. Sarosh, S. A. Parah, B. A. Malik, M. Hijji, and K. Muhammad, "Real-time medical data security solution for smart healthcare," *IEEE Trans. Ind. Inform.*, vol. 19, no. 7, pp. 8137–8147, Jul. 2023.
[4] R. Zhao, Y. Zhang, R. Lan, Z. Hua, and Y. Xiang, "Heterogeneous and customized cost-efficient reversible image degradation for green IoT," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2630–2645, Feb. 2023.
[5] K.-K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for Industrial Internet-of-Things: Research challenges and opportunities," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3567–3569, Aug. 2018.
[6] M. Gabr et al., "Application of DNA coding, the Lorenz differential equations and a variation of the logistic map in a multi-stage cryptosystem," *Symmetry*, vol. 14, no. 12, 2022, Art. no. 2559. [Online]. Available: https://www.mdpi.com/2073-8994/14/12/2559
[7] M. Alawida, "A novel chaos-based permutation for image encryption," *J. King Saud Univ.- Comput. Inf. Sci.*, vol. 35, no. 6, 2023, Art. no. 101595.
[8] W. Feng et al., "Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption," *Expert Syst. Appl.*, vol. 246, 2024, Art. no. 123190. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424000551
[9] N. N. Hurrah, S. A. Parah, J. A. Sheikh, F. Al-Turjman, and K. Muhammad, "Secure data transmission framework for confidentiality in IoTs," *Ad Hoc Netw.*, vol. 95, 2019, Art. no. 101989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157087051930424X
[10] Y. Sha, J. Mou, S. Banerjee, and Y. Zhang, "Exploiting flexible and secure cryptographic technique for multi-dimensional image based on graph data structure and three-input majority gate," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 3835–3846, Mar. 2024.
[11] Z. Gu et al., "IEPSBP: A cost-efficient image encryption algorithm based on parallel chaotic system for green IoT," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 89–106, Mar. 2022.
[12] J. Feng, J. Wang, Y. Zhu, and K. Han, "A hybrid chaotic encryption ASIC with dynamic precision for Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1148–1163, Jan. 2024.
[13] H. Wen and Y. Lin, "Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding," *Expert Syst. Appl.*, vol. 237, 2024, Art. no. 121514.

[14] W. Feng, Y. He, H. Li, and C. Li, "Cryptanalysis and improvement of the image encryption scheme based on 2 D logistic-adjusted-sine map," *IEEE Access*, vol. 7, pp. 12584–12597, 2019.
[15] H. Wen and Y. Lin, "Cryptanalyzing an image cipher using multiple chaos and dna operations," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 7, 2023, Art. no. 101612.
[16] V. P. Gupta, "Smart sensors and Industrial IoT (IIoT): A driver of the growth of industry 4.0," *Smart Sensors Ind. Internet Things: Challenges, Solutions Appl.*, pp. 37–49, 2021.
[17] W. Z. Khan, M. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Elect. Eng.*, vol. 81, 2020, Art. no. 106522.
[18] M. SaberiKamarposhti, D. Mohammad, M. S. M. Rahim, and M. Yaghobi, "Using 3-cell chaotic map for image encryption based on biological operations," *Nonlinear Dyn.*, vol. 75, pp. 407–416, 2014.
[19] H. Diab, "An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations," *IEEE Access*, vol. 6, pp. 42227–42244, 2018.
[20] M. Alawida et al., "A new hash function based on chaotic maps and deterministic finite state automata," *IEEE Access*, vol. 8, pp. 113163–113174, 2020.
[21] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.
[22] L. Huang, S. Cai, X. Xiong, and M. Xiao, "On symmetric color image encryption system with permutation-diffusion simultaneous operation," *Opt. Lasers Eng.*, vol. 115, pp. 7–20, Apr. 2019.
[23] N. A. Azam, J. Zhu, U. Hayat, and A. Shurbevski, "Towards provably secure asymmetric image encryption schemes," *Inf. Sci.*, vol. 631, pp. 164–184, 2023.
[24] S. Zhou, X. Wang, and Y. Zhang, "Novel image encryption scheme based on chaotic signals with finite-precision error," *Inf. Sci.*, vol. 621, pp. 782–798, 2023.
[25] W. Feng et al., "Exploiting newly designed fractional-order 3D Lorenz chaotic system and 2D discrete polynomial hyper-chaotic map for high-performance multi-image encryption," *Fractal Fractional*, vol. 7, no. 12, 2023, Art. no. 887.
[26] "The USC-SIPI image database," [Online]. Available: http://sipi.usc.edu/database/database.php
[27] U. Erkan, A. Toktas, F. Toktas, and F. Alenezi, "2D e-map for image encryption," *Inf. Sci.*, vol. 589, pp. 770–789, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521013475
[28] L. Teng, X. Wang, and Y. Xian, "Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion," *Inf. Sci.*, vol. 605, pp. 71–85, 2022.
[29] S. Gao et al., "Asynchronous updating Boolean network encryption algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4388–4400, Aug. 2023.
[30] W. Feng, X. Zhao, J. Zhang, Z. Qin, J. Zhang, and Y. He, "Image encryption algorithm based on plane-level image filtering and discrete logarithmic transform," *Mathematics*, vol. 10, no. 15, 2022, Art. no. 2751.



Moatsum Alawida (Member, IEEE) received the Ph.D. degree in computer science with a specialization in cybersecurity (cryptography) from the School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia, in 2020.

He is an Assistant Professor of cybersecurity with Abu Dhabi University, Abu Dhabi, UAE. With a prolific academic career, he has authored or coauthored more than 45 articles in high-impact factor journals. Notably, in the years 2021, 2022, and 2023, he achieved recognition among the top 2% of scientists worldwide. His diverse research interests include chaotic systems, applications of chaos, multimedia security, drone security blockchain, cybersecurity, quantum-based cryptography, and traditional cryptography methods.

Dr. Alawida was the Referee for esteemed journals, including IEEE, Elsevier, Springer, MDPI, and various conferences beyond his research contributions.