A Lightweight Cryptographic Solution for Enhanced Image Security

Mohammad Alnabhan
Department of Cybersecurity, School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan
m.alnabhan@psut.edu.jo

Ali El-Qasass
Department of Cybersecurity, School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan
ali20218110@std.psut.edu.jo

Mohammad Atoum
Computer Science Department, King Abdullah II School of Information Technology, University of Jordan, Amman, Jordan
m.atoum@ju.edu.jo

Qasem Abu Al-Haija
Department of Cybersecurity, Faculty of Computer & Information Technology, Jordan University of Science and Technology, Irbid, Jordan
qsabuhaija@just.edu.jo (corresponding author)

Ahmad Habboush
Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan
ah.habboush@ammanu.edu.jo

Received: 24 April 2025 | Revised: 25 June 2025 and 13 July 2025 | Accepted: 16 July 2025

 DOI: https://doi.org/10.48084/etasr.11707

ABSTRACT

Protecting the confidentiality of images is considered critical for all industries. Encryption stands out as the foremost protective strategy for image security. However, various image dimensions and quality concerns challenge conventional encryption methods in balancing the security objective, substantial computational resources, and image quality. This paper presents a lightweight cryptographic approach designed to match the robustness of traditional encryption schemes while minimizing the computational and time requirements to be more suitable for real-time applications and IoT environments. The proposed approach extracts the three Most Significant Bit (MSB) planes from the original image. Then, it applies a permutation on each bit plane using a chaotic logistic map to generate the key used as new indices for effective permutation of pixel locations. A set of performance metrics, such as the Correlation Coefficient (CC), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and CPU and RAM utilization, is used in the evaluation process. The results indicate that the proposed approach outperforms the traditional Advanced Encryption Standard (AES) in terms of computational efficiency and overhead. A 2% reduction in CPU and RAM utilization and a 96% decrease in extraction time were achieved while maintaining comparable security based on MSE, PSNR, and CC to AES. This improvement in efficiency is achieved while maintaining an acceptable level of security, which makes it more suitable for resource-constrained IoT devices.

Keywords-cryptography; histogram; Coefficient Correlation (CC); Mean Square Error (MSE); Peak Signal-to-Noise Ratio (PSNR); Advanced Encryption Standard (AES)

I. INTRODUCTION

The increasing reliance on the internet for data transmission has led to various cyberattacks targeting sensitive information. Images are one of the most widely transmitted data types on the Internet. Encryption can be considered one of the most important methods for protecting multimedia [1]. Conventional image encryption techniques focus on the frequency and space domains. Space domain-based image encryption has certain benefits in ensuring the integrity of the fused image and has a low computational cost [2]; however, the final image produced often lacks privacy due to poor encryption [3]. Scrambling is an important and simple technique in protecting images, which involves permuting pixel or bit values within a bit plane. Scrambling aims to transform an image into what appears to be meaningless noise, breaking the high correlation typically found between adjacent pixels [4].

Chaos is used to describe unpredictable behavior in nonlinear systems. Chaotic logistic maps are used to create pseudo-random sequences. Chaos and unpredictability can be introduced using the logistic map equation [5]:

$$xn + 1 = r \cdot x_n \cdot (1 - x_n) \quad (1)$$

Although traditional encryption techniques, such as AES, offer strong security, their computational complexity, due to multiple rounds of encryption and decryption operations, makes them unsuitable for resource-constrained IoT devices. Recent studies have explored lightweight cryptographic techniques, such as those based on chaotic maps and DNA-inspired algorithms. However, these methods may sometimes provide a different level of security than traditional techniques. However, lightweight techniques should provide high security.

To bridge this gap, this study presents a novel lightweight cryptographic approach focused on image confidentiality, ensuring a balance between security and computational overhead, and improved energy efficiency. The proposed approach leverages advanced techniques, such as chaotic maps and bit-plane manipulation, to enhance the security and efficiency of image encryption. Unlike typical encryption approaches that modify pixel values directly, this method uses only the three Most Significant Bit (MSB) planes for permutation. This preserves image quality while disrupting pixel correlation and allows for faster computation. The bit-plane selection balances security strength and minimal processing overhead, making it ideal for IoT scenarios.

Several recent studies have proposed lightweight image encryption methods that focus on chaotic maps and key permutation to improve efficiency. In [6], a multi-key AES-based encryption method was introduced. In [7], chaotic maps were combined with the Discrete Wavelet Transform (DWT) to increase randomness and performance. In [5], logistic map-based key generation was used to achieve strong statistical encryption properties. This highlights the ongoing pursuit of lightweight encryption, but most studies rely on pixel modification or complex transforms. The main contributions of this study can be summarized as follows:

- Uses a new lightweight approach for image encryption.
- Minimizes the process and rounds of encryption techniques.
- Uses a chaotic logistic map to generate a random permutation of the extracted bit planes of the image pixels.

II. RELATED WORK

Image protection has gained great attention, particularly in sensitive areas such as military and health applications. The aim is to ensure that the images remain understandable only to authorized users. Researchers have explored various advanced methods to protect images. In [6], an encryption approach utilized five different types of keys generated randomly or provided by users. These keys varied in size and generation method: Key1 was an 8-byte random key, Key2 consisted of two 4-byte random parts, Key3 was a 4-byte user-input key, Key4 combined a 4-byte random key with a 4-byte user-input key, and Key5 was an 8-byte user-input key. This study compared JPEG image encryption with symmetric encryption algorithms, such as DES and AES. Metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) were used to assess performance, and the results showed that the use of Key2 and Key3 achieved the best MSE results.

In [7], chaotic maps were combined with DWT. This approach generated a random vector through a chaotic equation, which creates permutations applied to the image in the frequency domain. The image is then returned to the spatial domain, and an XOR operation with a chaotic key matrix is performed. This method was evaluated using statistical matrices, showing improved security performance. In [8], a GUI application in Java employed AES encryption with a 128-bit key for protecting images across various formats during transmission over insecure channels. This study highlighted the advantages of AES encryption, particularly its suitability for sensitive applications due to its support for multiple key lengths (128, 192, and 256 bits).

In [9], the focus was on dual scrambling techniques for image encryption. The standard zigzag scrambling algorithm was combined with logistic mapping to enhance security. This method minimized the similarity between the original and scrambled images by altering the pixel order, resulting in robust security and low complexity. Evaluations using histograms, PSNR, and MSE confirmed the effectiveness of this approach. In [10], a high-throughput 256-bit AES cipher was developed for digital images, focusing on peer-to-peer communication. The images were first converted to grayscale and resized before being encrypted in 4×4 pixel blocks. This method achieved a PSNR of 61 dB and an MSE of 0.0030, demonstrating the efficacy of AES 256 compared to AES 128 and 192.

In [11], RSA encryption was explored by dividing images into 2×2 blocks, converting them to vectors and decimal numbers, and then applying RSA encryption with a public key. This approach was evaluated using visual and numeric metrics such as histogram, Correlation Coefficient (CC), PSNR, and information entropy, showing effective protection against statistical attacks and high disorder in encrypted images. In [12], encryption and steganography were combined by proposing a method using AES 128 CBC to protect both the

images and embedded secret information. This approach was extended to protecting PDF files during transmission over insecure networks. A comparative study in [13] assessed the symmetric AES and asymmetric RSA algorithms for image encryption. Metrics such as CC and histogram analysis showed that AES outperforms RSA in image protection. In [14], various encryption algorithms, such as AES, RSA, and Elliptic Curve Cryptography (ECC), were evaluated, finding that AES and RSA were faster, while ECC offered the highest security, highlighting the trade-offs between performance and security.

In [5], a chaotic map-based key generation method was introduced for high-security image encryption. Using random initial values to generate a unique key significantly complicates key detection. In [15], a novel approach utilized convolutional neural networks for double-image encryption. This method employed a chaotic sequence as the convolution kernel, which governed the scrambling process within the encryption algorithm. The approach was evaluated using MSE and PSNR metrics, demonstrating the potential of combining neural networks with encryption techniques.

In [16], a lightweight image encryption algorithm was based on logistic maps, permutations, and the AES S-box, showing improvements in speed and security compared to conventional AES methods. In [17], various lightweight cryptographic algorithms were investigated based on block and stream ciphers, providing a detailed analysis of their performance metrics in the context of applicability to the IoT environment. In [18], a lightweight image encryption algorithm was based on a modified NonLinear Feedback Shift Register (NLFSR). This algorithm used pseudo-random sequences for the image permutation and diffusion operations. The evaluation results confirmed its robustness and efficiency in resource-constrained environments.

According to previous studies, one of the main challenges in the field of image protection is the complexity of the encryption and decryption processes, particularly due to the multiple rounds involved. Additionally, most existing studies did not test their methods against compression, and the focus was on grayscale images. In contrast, the proposed approach simplifies the encryption rounds by only permuting the location of the pixels, eliminating the need for multiple rounds. Several compression scenarios were used for evaluation under different quality metrics. Furthermore, the proposed method successfully handles RGB images, demonstrating greater adaptability. The security of this approach was further enhanced through the combined use of bit-plane techniques, chaotic logistic maps, and permutation concepts.

III. METHODOLOGY

This study aimed to design a new approach for protecting sensitive images to achieve high confidentiality with decreased overhead. This approach consists of two phases: (i) slicing the image by taking the three most significant bit planes, and (ii) using a chaotic logistic map to create a random sequence for the permutation of the planes. Figure 1 illustrates the process flow of the proposed approach. The metrics used to evaluate the proposed approach were the following:

- Correlation Coefficient (CC) quantifies the strength of a statistical relationship between two variables (adjacent pixels) [7]:

$$r\frac{nx(\sum(X,Y)-(\sum(X)x\sum(Y))}{\sqrt{nx\sum(X^2)-\sum(X^2)x(nx\sum Y^2)-\sum(Y^2))}} \quad (2)$$

- A histogram is one of the most common visual methods to represent each color in an image, which shows how many pixels correspond to each color and its intensity. A histogram is supposed to exhibit a uniform distribution for a cipher image.

- MSE is used to find the difference between the plain image A and the cipher image B, where the lowest value is preferable in encryption. It can be calculated using [9]:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[A(i,j) - B(i,j)]^2 \quad (3)$$

- Peak Signal to Noise Ratio (PSNR) evaluates the visibility of each image and depends on the MSE as follows [9]:

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (4)$$

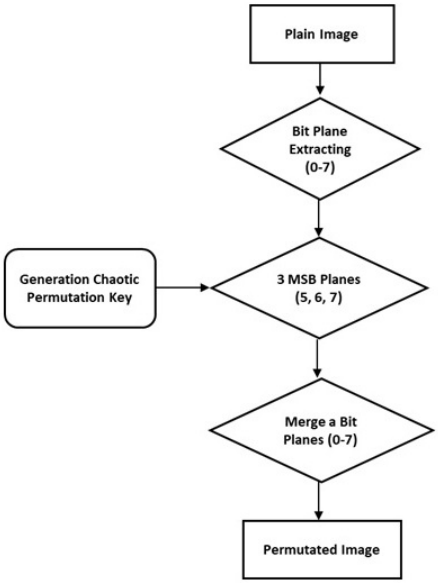

Fig. 1. Proposed cryptographic approach.

The following control parameters generate chaotic values from a logistic map: x_0, r, where x_0 is the initial value $\in(0,1)$ and $r \in (3.7, 4)$. If $x_0 = 0.5$ and $r = 3.9$, then the output values will be $x_1 = 3.9 * 0.5 * (1 - 0.5) = 0.975$, $x_2 = 3.9 * 0.975 * (1 - 0.975) \approx 0.09506$, $x_3 = 3.9 * 0.09506 * (1 - 0.09506) \approx 0.3353$, etc., where x values are generated according to the size of each bit plane. As the value of $X_n \in (0,1)$, different methods are used to make the numbers in between the range of the bit plane size. Considering the size is $H * W$ (H : height, W : width), then the formula $(X_n * 1016)\ mod\ H * W$ is used to generate values $\in (0, (H * W) - 1)$ which will be used to permutate the pixels of each bit plane. In addition, the Python function argot can be applied to the X_n values to sort each bit plane pixel according to the chaotic key.

The proposed approach was implemented using Anaconda/Spyder (Python 3.11), with Lena's image encrypted by AES, acting as a benchmark for evaluation. Initially, grayscale images were used in the early testing phases because of their lower complexity and to facilitate proof-of-concept development. However, the final implementation was extended to RGB images to ensure broader applicability. The adaptation required managing each color channel independently and applying the same permutation logic to the three MSB planes in each channel. Adapting the proposed method to RGB images introduces several challenges compared to grayscale. Each color channel must be treated separately, resulting in three times the number of bit planes to process and permute, and increasing memory consumption and processing time. In addition, maintaining the integrity of the image requires synchronization between channels to avoid perceptual artifacts. These trade-offs were managed by optimizing memory access and reusing permutation sequences when feasible. One limitation identified is the scalability of the approach for high-resolution images. Since each pixel in each MSB plane must be permuted, larger image sizes increase memory usage and runtime. Further optimization or parallelization strategies are necessary to support real-time processing at higher resolutions.

The first visual metric applied was a histogram, where the encryption produces a histogram shape that is uniform and different from the plain image histogram, as shown in Figure 2, which is plotted for the grayscale image, not the RGB colored image.

The CC representing the correlations between the neighbor's pixels and in the plain image is near 1, while after a good encryption, it will be near 0. Figure 6(a) illustrates the relationship between neighboring pixels in the plain image, with a CC close to 1, indicating a strong correlation. In contrast, Figure 6(b) shows the relationship between neighboring pixels after encryption, with the correlation coefficient reduced to around 0, indicating a lack of correlation.

MSE is also a quantitative metric for evaluating pixel value changes after encryption, where the largest value means good encryption. In addition, PSNR evaluates the cipher quality, where the lowest value indicates strong encryption, which makes the image noisy after encryption. AES results in Table I show that the encryption process produces a high MSE between the plain and cipher images, indicating a significant difference. Additionally, MSE between the plain and recovered images is close to 0, meaning that the recovered image is almost identical to the original plain image.

TABLE I. AES ALGORITHM RESULTS

AES Algorithm	Results
MSE (Lena, Cipher)	7789.9
MSE (Lena, Decrypt image)	0.0
PSNR (Lena, Cipher)	9.21554

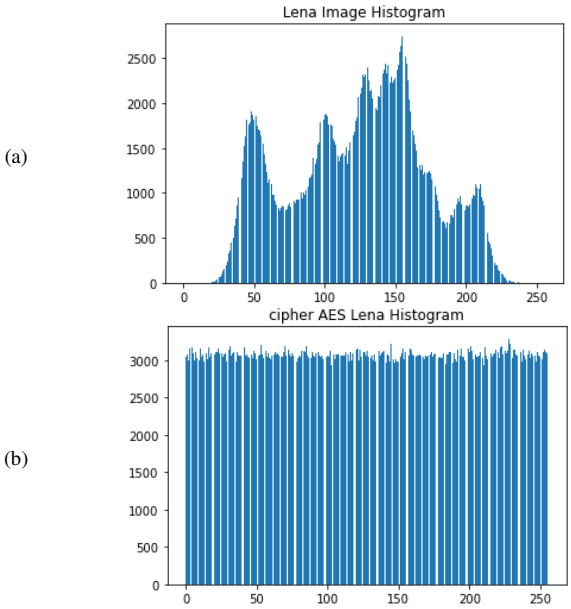

Fig. 2. (a) Plain image histogram, (b) cipher image histogram using AES.

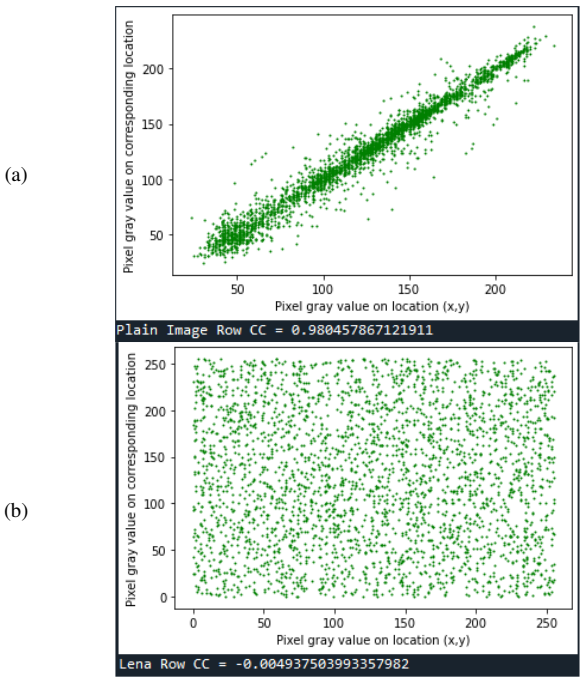

Fig. 3. (a) CC for plain image, (b) CC for cipher image.

A. *Proposed Approach*

As described earlier, the proposed approach combines bit plane slicing and a chaotic logistic map. Equation (1) generates a random key for the permutation of the 3 MSB planes. The process of permutating the plain image includes the following steps:

- Load the plain image.
- Extract eight (0-7) bit planes of the image.

- Use only three MSB planes (5,6,7).
- Use the chaotic logistic map (1) to generate the random permuted key. The permutation is applied to each of the 3 MSB planes. Finally, merge all planes to get the permutated image sent to the destination part.

The pseudocode of the image permutation function can be described as follows.

```
Algorithm 1: Image Permutation
Input: plain image, chaotic key (r, x),
  planes to permute
Output: permuted image

Function: permutate_image(original_image,
  chaotic_key, permuted_planes_indices):
  Convert the original image to an array
  format
  Initialize arrays for planes and
  permuted_planes
  For each color channel (Red, Green,
  Blue):
    Extract bit planes
    Generate a permutation sequence using
    a logistic map with a chaotic key
    Permute specified bit planes using a
    sequence
    Store permuted planes
  Merge the permuted planes with the
  original ones for each color channel
  Combine color channels into one image
  Return the merged image

Function: logistic_map(x, r, size):
  Generate and return a sequence using a
  logistic map for a given size

Function: permute_planes(planes, indices,
  permutation_sequence):
  Flatten, permute, and reshape specified
  planes using the permutation sequence
  Return permuted planes

Function: merge_planes(planes,
permuted_planes, permuted_indices):
  Merge the permuted and original planes
  into the final image
  Return the merged image
```

The process of recovering the plain image includes the following steps:

- Load the permutated image.
- Extract the eight-bit planes (0-7).
- Use the 3 MSB to de-permutate the original plain 3 MSB using the chaotic key in the permutated process.
- Merge all planes to recover the original plain image.

The pseudocode of the image recovery function can be described as follows.

```
Algorithm 2: Image Recovery
  Input: permutated image, chaotic key (r,
  x), planes to permute
Output: original (plain) image

Function: reconstruct_original
  (merged_image, original_planes,
  permuted_indices):
  Initialize reconstructed image array
  For each bit plane index (0 to 7):
    If an index is in permuted_indices:
      Use the original plane
    Otherwise:
      Extract and use the corresponding
      plane from merged_image
  Return the reconstructed image

Function: Main
  Load the original image and convert it
  to an RGB array
  Initialize arrays for planes and
  permuted_planes for each color channel
  For each color channel (Red, Green,
  Blue):
    Extract bit planes
    Generate a permutation sequence using
    a logistic map with a chaotic key
    Permute specified bit planes using a
    sequence
    Store both original and permuted
    planes
    Merge the permuted and unpermuted
    planes into one merged image
    Reconstruct the original image from a
    merged image using
    reconstruct_original
    Display the original and merged images
```

B. Evaluation Results

The Google Colab environment and Anaconda/Spyder package were used to run and evaluate the proposed method on the colored (RGB) Lena's image. The histogram analysis showed an insignificant difference between the original and permuted images, as shown in Figure 4(a,b). This shows that the proposed approach only alters the location of the pixels, not their values, where the histogram of the permutated image is close to a uniform distribution, which means good efficiency.

As shown in Figure 5, the proposed approach produced a CC for the permutated image similar to any traditional encryption technique. Producing CC near zero after the permutation of the plain image indicates that the proposed approach successfully changed the correlation between the neighboring pixels. Figure 6 shows the permutated image.

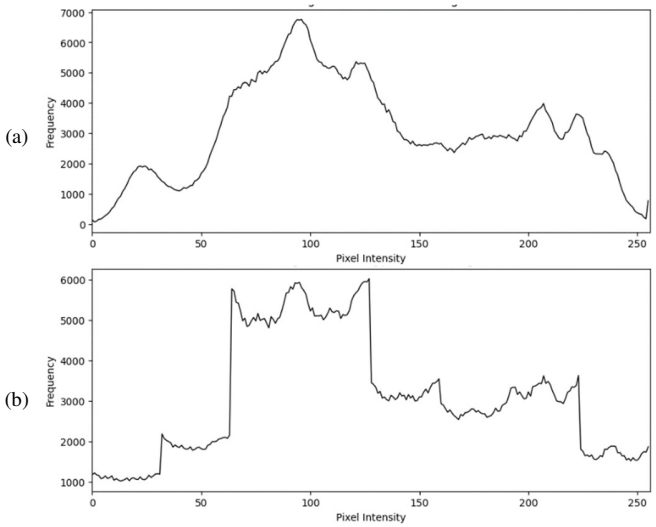

Fig. 4. (a) Histogram for plain Lena image, (b) Histogram for permutated image (Encrypted Lena image).

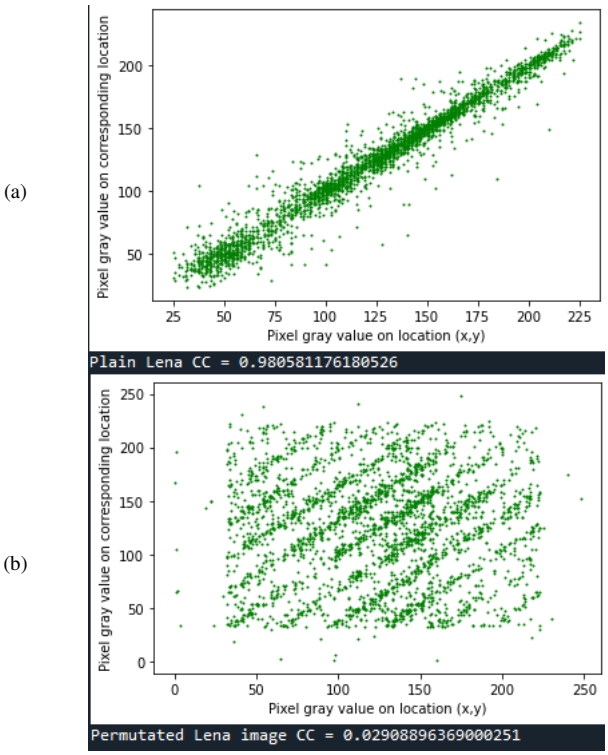

Fig. 5. (a) CC for plain image, (b) CC for permutated image.

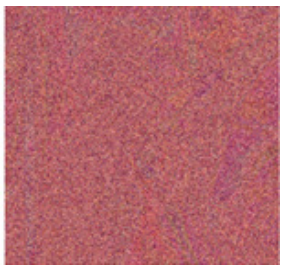

Fig. 6. Permutated image.

The results for MSE and PSNR were very similar to those when using the AES algorithm, showing that the proposed approach produced a permutated image near the encrypted one. According to Table II, the proposed approach yielded results comparable to those of the AES algorithm, demonstrating that it can be used with fewer processes and rounds while maintaining good efficiency.

TABLE II. RESULTS OF THE PROPOSED APPROACH

Metrics	Results
MSE between Lena & reconstructed image	0.0
MAP	4491.6
PSNR	11.61

IV. RESULTS AND ANALYSIS

Python was used to implement and test the proposed approach, along with multiple libraries (matplotlib for visualizations, PIL for images, NumPy for dealing with numerical data, math for mathematical functions, time for dealing with time-related functions, and psutil for system utilization). Functions were defined using the def command for extract plane, logistic map, permutate plane, merge planes, monitor resources, and extract the original from the merged image. Common metrics were also used to evaluate the proposed approach, such as Histogram, PSNR, MSE, and CC. Table III presents the performance results of conventional AES encryption and the proposed approach, both applied to the same RGB Lena image for consistent comparison. Conventional AES offers higher MSE, PSNR, CC, and encryption time, as it functions in normal encryption operations, incorporating complex computations and multiple rounds that alter pixel values. In contrast, the proposed approach relies on simpler operations, such as slicing and permutation, and delivers outcomes that, while acceptable, only produce a scrambled image devoid of any discernible meaning. Such images cannot be decoded or reconstructed through Human Vision System (HVS) analysis or statistical methods, unless the specific algorithm and its parameters are known. Hence, the proposed approach outperforms conventional AES in terms of execution time, CPU usage, and RAM usage. Given this lower computational overhead, the proposed method suits applications and fields with limited resources, such as the IoT sector and mobile devices.

TABLE III. COMPARISON OF EVALUATION METRICS

Metrics	AES encryption	Proposed approach
MSE	7789	4491
PSNR	9.2	11.6
CC	0.001	0.020
Histogram	Uniformly-distributed	Semi-uniformly-distributed
Encryption time	0.0205	0.1829
Extraction time	0.0347	0.0013
CPU used	5%	3.5%
RAM used	9.6%	7.8%

Table IV compares the results of the proposed method with other studies. The proposed approach achieved an MSE of 4491.6 between the original and the permutated images, which is lower than the MSE values reported in [6, 15]. The proposed

approach achieved better MSE results than [7], considering that the image permutation was conducted without encryption. The average PSNR value achieved by the proposed approach was 11.6, comparable to those reported in [6, 7, 9, 11]. The MSE and PSNR values of the proposed approach were achieved by permuting image pixels without changing the pixel values, as in traditional encryption methods described in previous research.

TABLE IV. COMPARISON WITH EXISTING MODELS

Ref.	Year	Method	MSE	PSNR
[6]	2022	Multi-key AES-based symmetric encryption for images	10381	7.96
[7]	2022	Chaotic map & DWT	379	12.65
[9]	2019	Encryption algorithm based on double scrambling	Not evaluated	7.6
[10]	2023	256-bit AES	0.003 between plain and recovered image	61 for the recovered image
[11]	2022	RSA-based encryption	Nor evaluated	9.22
[15]	2021	Using NN	6988	9.6

V. CONCLUSIONS AND FUTURE WORK

This paper presents a new approach to lightweight image encryption, which integrates the bit plane concept with the chaotic concept, where the permutation process is based on a random value generated by a mathematical equation with an initial value. Using the bit plane method provides an additional layer of security and ambiguity, resulting in more strength and robustness in encrypted images. The proposed approach was evaluated using an experimental setup, and the results achieved were compared with conventional AES image encryption and other methods. The proposed approach showed its lightweight overhead compared to other methods, confirming its suitability for implementation in resource-constrained applications.

However, the proposed method has certain limitations. Since encryption relies on bit-plane permutation without altering pixel values, it may be less resilient to more advanced cryptographic attacks. Additionally, the scalability of the method for high-resolution or large datasets has not been fully explored and remains an area for further investigation. For example, the average CPU and RAM utilization during image encryption was 3.5% and 7.8% for the proposed approach, compared to 5% and 9.6% for conventional AES, respectively.

In the future, the proposed approach will be applied to the frequency domain instead of the spatial domain to add more layers of security. Another extension can deal with all bit planes instead of only 3 MSB, and then apply another chaotic equation instead of the logistic map. In addition, the proposed approach will be evaluated against advanced and persistent security attacks. There is also a plan to incorporate additional encryption layers, such as lightweight substitution or diffusion stages, to further enhance security and make the method suitable for broader, high-risk applications. In addition, key management remains a critical consideration. Since encryption relies on chaotic keys generated from initial parameters (x_0 and r), secure distribution and synchronization of these values between the sender and the receiver must be ensured to prevent compromise.

REFERENCES

[1] V. K. P. Kalubandi, H. Vaddi, V. Ramineni, and A. Loganathan, "A novel image encryption algorithm using AES and visual cryptography," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India, Oct. 2016, pp. 808–813, https://doi.org/10.1109/NGCT.2016.7877521.

[2] A. Kadir, M. Aili, and M. Sattar, "Color image encryption scheme using coupled hyper chaotic system with multiple impulse injections," *Optik*, vol. 129, pp. 231–238, Jan. 2017, https://doi.org/10.1016/j.ijleo.2016.10.036.

[3] L. Liu and S. Miao, "A new image encryption algorithm based on logistic chaotic map with varying parameter," *SpringerPlus*, vol. 5, no. 1, Mar. 2016, Art. no. 289, https://doi.org/10.1186/s40064-016-1959-1.

[4] B. Mondal, "Cryptographic Image Scrambling Techniques," in *Cryptographic and Information Security*, 1st ed., S. Ramakrishnan, Ed. Boca Raton, FL, USA: CRC Press, 2018, pp. 37–65.

[5] Z. Mu and H. Liu, "Research on digital media image encryption algorithm based on Logistic chaotic map," in *2020 International Conference on Robots & Intelligent System (ICRIS)*, Sanya, China, Nov. 2020, pp. 108–111, https://doi.org/10.1109/ICRIS52159.2020.00035.

[6] G. Spasova and M. Karova, "A New Secure Image Encryption Model Based on Symmetric Key," in *2021 International Conference on Biomedical Innovations and Applications (BIA)*, Varna, Bulgaria, Jun. 2022, pp. 107–110, https://doi.org/10.1109/BIA52594.2022.9831258.

[7] A. Shafique and J. Ahmed, "A Color Image Encryption Algorithm Based on Chaotic Map and Discrete Wavelet Transform," in *2022 Global Conference on Wireless and Optical Technologies (GCWOT)*, Malaga, Spain, Feb. 2022, pp. 1–5, https://doi.org/10.1109/GCWOT53057.2022.9772906.

[8] S. Saudagar *et al.*, "Image Encryption based on Advanced Encryption Standard (AES)," in *2023 International Conference for Advancement in Technology (ICONAT)*, Goa, India, Jan. 2023, pp. 1–4, https://doi.org/10.1109/ICONAT57137.2023.10080243.

[9] H. Wang, Q. Wang, L. Yu, and J. Zhao, "Image Encryption Algorithm Based on Double Scrambling," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, Tianjin, China, Aug. 2019, pp. 2201–2205, https://doi.org/10.1109/ICMA.2019.8816275.

[10] A. Hennache, M. L. Hennache, and S. M. A. Ghaly, "Improving the RSA Encryption for Images by Introducing DNA Sequence Encoding," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17786–17791, Dec. 2024, https://doi.org/10.48084/etasr.8557.

[11] S. J. Edan, M. N. Rasoul, and A. A. Aljarrah, "RSA-based Encryption Algorithm for Digital Images," in *2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)*, Basrah, Iraq, Sep. 2022, pp. 303–308, https://doi.org/10.1109/IICCIT55816.2022.10010627.

[12] N. Sofian, A. Wicaksana, and S. Hansun, "LSB Steganography and AES Encryption for Multiple PDF Documents," in *2019 5th International Conference on New Media Studies (CONMEDIA)*, Bali, Indonesia, Oct. 2019, pp. 100–105, https://doi.org/10.1109/CONMEDIA46929.2019.8981842.

[13] E. S. Atwal and U. Kumar, "A Comparative Analysis of Different Encryption Algorithms: RSA, AES, DSS for Data Security." Preprints.org, Apr. 26, 2021, https://doi.org/10.20944/preprints202104.0673.v1.

[14] A. Chaouch, B. Bouallegue, and O. Bouraoui, "Software application for simulation-based AES, RSA and elliptic-curve algorithms," in *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Monastir, Tunisia, Mar. 2016, pp. 77–82, https://doi.org/10.1109/ATSIP.2016.7523051.

[15] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, "Double image encryption algorithm based on neural network and chaos," *Chaos, Solitons & Fractals*, vol. 152, Nov. 2021, Art. no. 111318, https://doi.org/10.1016/j.chaos.2021.111318.

[16] Y. Alghamdi, A. Munir, and J. Ahmad, "A Lightweight Image Encryption Algorithm Based on Chaotic Map and Random Substitution," *Entropy*, vol. 24, no. 10, Sep. 2022, Art. no. 1344, https://doi.org/10.3390/e24101344.

[17] S. A. Abead and N. H. M. Ali, "Lightweight Block and Stream Cipher Algorithm: A Review," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 5, no. 2, pp. 860–874, Jun. 2024, https://doi.org/10.37385/jaets.v5i2.3966.

[18] P. Kumari and B. Mondal, "Lightweight encryption scheme based on a new NLFSR," *Multimedia Tools and Applications*, vol. 83, no. 24, pp. 64919–64943, Jul. 2024, https://doi.org/10.1007/s11042-024-18222-y.