Available online at www.garmian.edu.krd

Passer Journal of Basic and Applied Sciences

Passer 7 (Issue 2) (2025) 707-716

Original Article

Parallel AES Image Encryption with Logistic Map

Mohammed Dheyaaldeen Taha* [1]

[1] *Kirkuk Education Directorate, Kirkuk 36001, Iraq.*

Received 02 March 2025; revised 01 May 2025;
accepted 09 June 2025; available online 29 July 2025

DOI: 10.24271/PSR.2025.509863.1986

ABSTRACT

The security of multimedia data, especially photographs, has grown in importance as digital communication has expanded quickly. The specialty area of information security known as "image cryptography" is concerned with securing image data using encryption methods to stop illegal access, alteration, or leaking. In contrast to text-based data, photos have a lot of information with strong pixel correlations and redundancy, which presents special potential and challenges for cryptographic applications. In this research, a logistic map-based chaotic system is combined with the Advanced Encryption Standard (AES) to provide a novel method for improving the security and efficiency of picture encryption. AES is implemented in parallel, enabling quicker encryption of high-resolution photos, and S-boxes and AES keys are dynamically generated using chaotic sequences drawn from the logistic map. It operates on blocks of the fixed size of 128 bits with a key size of 128, 192, or 128,192,256 bits and ten rounds are taken during the encryption and decryption processes. According to experimental results, the system is resilient to brute force and differential assaults, which makes it a viable option for the secure transmission of images in contemporary communication networks. The test results with a 32-byte key for the Flower image show that throughput, memory usage, speed-up, efficiency, and data processing factors are 333.3415 MB/s, 147.94 MB, 1.1306 s, 11.56, and 185486435.86 bytes/sec, respectively. The suggested approach performs better in terms of encrypted quality, random unpredictability, and processing when tested on both color and grayscale photos.

Keywords: S-Box, hybrid algorithm, image encryption, parallel AES, logistic map.

1 Introduction

The rapid growth of digital communication has heightened the need for secure and efficient methods to protect sensitive visual data, such as images, during transmission. Images are widely used in critical applications, including medical imaging, military systems, and multimedia platforms, making them a prime target for cyber threats[1,2]. Traditional encryption techniques, while effective for text-based data, often struggle to address the unique challenges posed by images, such as their large size, high redundancy, and specific structural characteristics. In order to get over these restrictions, scientists have looked into combining cryptographic algorithms with chaotic systems, which provide increased security due to characteristics like ergodicity, pseudo-randomness, and sensitivity to beginning conditions[1,3,4].

A popular symmetric encryption method renowned for its effectiveness and resilience is the Advanced Encryption Standard (AES). But because of its fixed key generation procedure and static substitution box (S-box), it may have flaws that leave it open to specific cryptographic assaults[5, 6]. This paper suggests a fresh solution to these problems by combining a chaotic logistic map with AES. Because the S-box and encryption key are dynamically generated using the logistic map, the encryption process is highly unpredictable and random[7]. In addition to enhancing AES security, its dynamic generation guarantees defense against attacks like differential and linear cryptanalysis [8].

Additionally, a parallel implementation of AES is provided to improve the encryption process' efficiency. Large image file encryption is greatly expedited by utilizing parallel computing techniques, which makes the approach appropriate for real-time applications[8]. For secure image encryption, a strong, scalable, and effective framework is created by combining chaotic systems for dynamic S-box and key creation with parallel AES execution.

[9]. This strategy responds to the increasing need for cutting-edge encryption methods that can guarantee current digital communication systems' performance and security. The suggested approach makes a significant contribution to the field of image encryption and gives a viable way to protect private visual information in a world that is becoming more networked by the day[9,10].

* Corresponding author
E-mail address msc.muhammed84@gmail.com (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

2 Related Work

Researchers are exploring using chaotic systems, specifically logistic maps, in cryptography due to their ability to generate pseudo-random sequences with high initial condition sensitivity.

Author in [11], in order to safeguard color images, this work proposes a color image encryption technique and uses the one-dimensional logistic map to construct an s-box. The proposed color picture encryption technique is based on the recently developed S-box. The new S-box satisfied the balanced, comprehensive, avalanche, and strict avalanche S-box test criteria. Furthermore, encryption performance analysis metrics are documented, including information entropy, correlation analysis, histogram analysis, and differential attack. The results show that the produced Lena image has information entropy, NPCR, and UACI values of 7.9972, 99.601%, and 33.5647%, respectively.

Author in [12], this study used a 1D logistic map to construct a novel S-Box based on a chaotic system. The new excellent S-Box was then built utilizing the hexa code values that were extracted from the chaotic sequence that the 1D logistic map created. To evaluate this idea, the S-Box test criteria—avalanche, balanced, completeness, rigorous avalanche, and inevitability were applied. Since the proposed S-Box passes all of these statistical criteria and has a sizable avalanche effect, the analytical results show that it can avoid many attacks. Because it just takes a few moments to produce, Substitution-Box can be used in a number of lightweight block ciphers.

Author in [13], the integration of the logistic map into AES security is a novel method that strengthens encryption, particularly in the face of new security threats. This integration significantly enhances metrics like Integrity Check, Degree of Diffusion, and Degree of Confusion.

An important area of attention has been parallel AES implementations to overcome the computational difficulties associated with image encryption, especially for high-resolution or real-time applications:

Author in [14], image encryption uses chaos-based concurrent algorithm, splitting images into threads and generating keys from hyper chaotic systems. Experimental results show ideal information entropy screening rate, faster results, and improved security.

Author in [15], the study examines how GPUs can speed up encryption procedures by utilizing the Advanced Encryption Standard (AES) algorithm. It was discovered that while GPU processing shortened processing times for large data, it was not required for low-size encryption. After reaching the maximum load value, the acceleration speed stays constant.

Author in [16], this work introduces two effective pipelined and parallel data encryption algorithms that minimize processing time. Large data sets benefit greatly from these techniques, which enable the encryption of several states at once without waiting for earlier operations.

3 Research Background

3. 1 AES Algorithm

A symmetric-key block cipher called the Advanced Encryption Standard (AES) is used to encrypt sensitive data. Three different key sizes are supported by AES, which was created in 2001 and runs on fixed-size blocks of 128 bits with key (128 ,192 ,256) bits. The algorithm's number of rounds throughout the encryption and decryption procedures is determined by the key size. By transforming the data block in a number of ways, AES operations like Sub Bytes, Shift Rows, Mix Columns, and Add Round Key strengthen and complicate the encryption. [5,6]. shown in "Figure 1" below

Modes of Operation: AES may encrypt data larger than a single block (128 bits) in a variety of ways. Typical modes include the following:

- **ECB (Electronic Codebook):** The same key is used to independently encrypt every textual block. Despite being straightforward, it is susceptible to pattern attacks.
- **Cipher Block Chaining (CBC):** This method offers greater security than ECB by Exoring each plaintext block with the preceding cipher text block prior to encryption.
- **GCM (Galois/Counter Mode) and CTR (Counter):** These modes provide improved security and performance, and GCM further offers authenticated encryption, which guarantees data integrity.

AES has strong security, efficiency, and flexibility, making it suitable for a wide range of applications. However, it has weaknesses in key management, vulnerabilities in certain modes, and potential quantum threats[5,6].
In conclusion, AES is a highly secure and efficient encryption algorithm widely adopted for securing sensitive data, but careful attention to key management and mode selection is necessary to ensure its security in real-world scenarios[13].

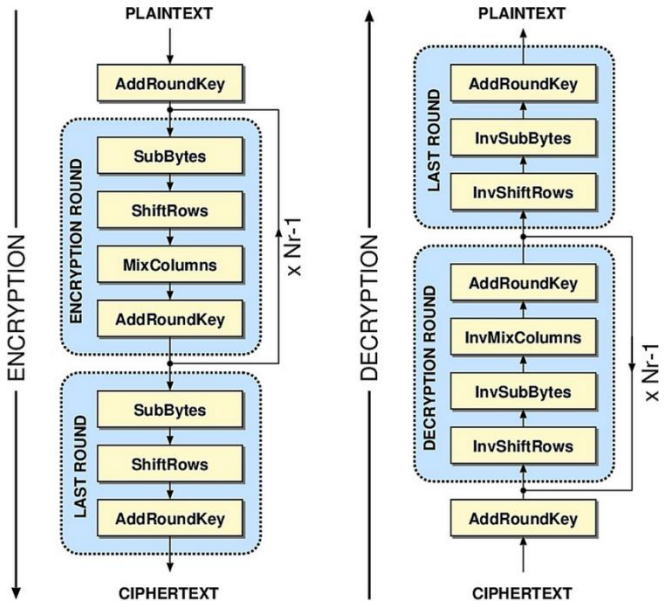


Figure 1: AES's Basic Structure [5,6].

3. 2 Logistic Map

In a deterministic system, chaotic behavior is a long-term, aperiodic behavior. The "butterfly effect" refers to how sensitive chaos-based systems are to beginning conditions. Chaos-exhibiting nonlinear dynamical discrete temporal systems are referred to as "chaotic maps." A chaotic map has the benefit of being deterministic. In order to boost security, many researchers exploit a chaotic system and combine it with the advantageous aspects of cypher cryptography, such as confusion and diffusion. Systems that require security, including block and stream ciphers, image encryption methods, etc., can be utilized with the chaos system. Table 1 displays several chaotic systems, the logistic map being one of the most widely employed. One-dimensional logistic maps are straightforward and have the potential to behave chaotically. In general, Eq. (1) can be used to mathematically represent one-dimensional logistic maps[11,12].

$$x_{i+1} = r * x_i * (1 - x_i) \quad (1)$$

where r is the system control parameter, x_0 is the initial state, and i is the number of repeats. The value of x_{i+1} is a number between (0-1) for all i, as shown in "Figure 2" below, however the value of the control parameter r falls within the interval (0,4) and produces better values when r is almost equal to 4. shown in "Figure 3" below that depending on the value of r, x displays a variety of behaviors. This work builds a new S-Box using the 1D logistic map method[11].

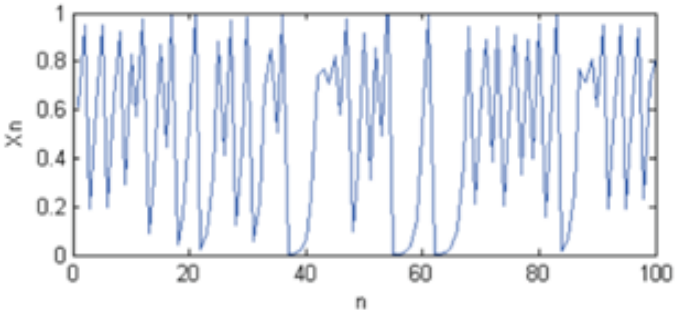


Figure 2: Iteration values and logistic chaotic behavior [11,12].

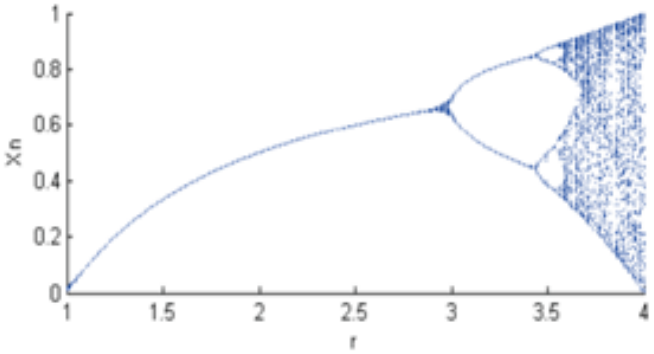


Figure 3: A logistic map bifurcation diagram [11,12].

4 Proposed Method

By using the logistic map to create the AES encryption key and the S-Box for the encryption process, we hope to improve on the conventional AES encryption by integrating chaos theory. This method makes use of parallelism and the logistic map's pseudo-randomness to enhance picture encryption's security and performance.

4. 1 Generating an AES key (128,192,256) bits and IV(128 bits) using a logistic map:

The AES key is constructed using the pseudo-random numerical sequence that is produced by the logistic map. An initial value x0 and a control parameter r, which regulates the key's complexity and randomness, are used to create this sequence. To fit the byte size of the AES key, the values produced by this recurrence are scaled to the interval [0, 255]. The first 16 values from the logistic map sequence are used to generate the key. These values are then transformed into bytes and used to set up the AES encryption.

4. 2 Constructing S-Box Using Logistic Maps:

The logistic map uses a sequence to create a custom S-Box, which is swapped with matching values in the AES replacement step. This disorganized behavior strengthens AES encryption's confusion feature and makes it more resistant to cryptographic attacks. (See Algorithm (4.1)).

ALGORITHM (4.1): CONSTRUCTING S-BOX USING LOGISTIC MAP 1D

INPUT Initial Input parameter (r=3.99, xi=0.59)

OUTPUT New S-Box [256]

BEGIN

STEP 1 Read initial parameter (r, x0)

STEP 2 Iterate Logistic Map Equation $x_{i+1} = r \times x_i \times (1-x_i)$

STEP 3 Generation Logistic Map

STEP 4 Scale to Byte Range (yn= [xn× 256]

STEP 5 Check for Duplicates

STEP 6 If Duplicates Found

Yes - Reinitialize or Adjust Sequence -iterate Logistic Map Equation(Step2)

No - Duplicates Create S-box

STEP 7 Validation Fails Reinitialize Parameters Input parameters (r, xi) (Step1)

Validation passes Output S-Box

END

4. 3 Parallel AES encryption & decryption

4. 3. 1 Parallel AES encryption:

- After loading and converting the image to be encrypted into a byte array, it is padded to satisfy the AES encryption block size specifications.
- We use Python's Thread Pool Executor for parallel processing in order to increase encryption performance and lower computational overhead. This speeds up the process, especially for large photos, by enabling us to divide the data into chunks and execute the substitution phase (using the modified S-Box) concurrently.

Following parallel replacement, CBC (Cipher Block Chaining) mode is used to apply the AES encryption process (using the produced key and the custom S-Box). This ensures that each encrypted data block depends on the one before it, improving security. Show in" Figure 4" below.

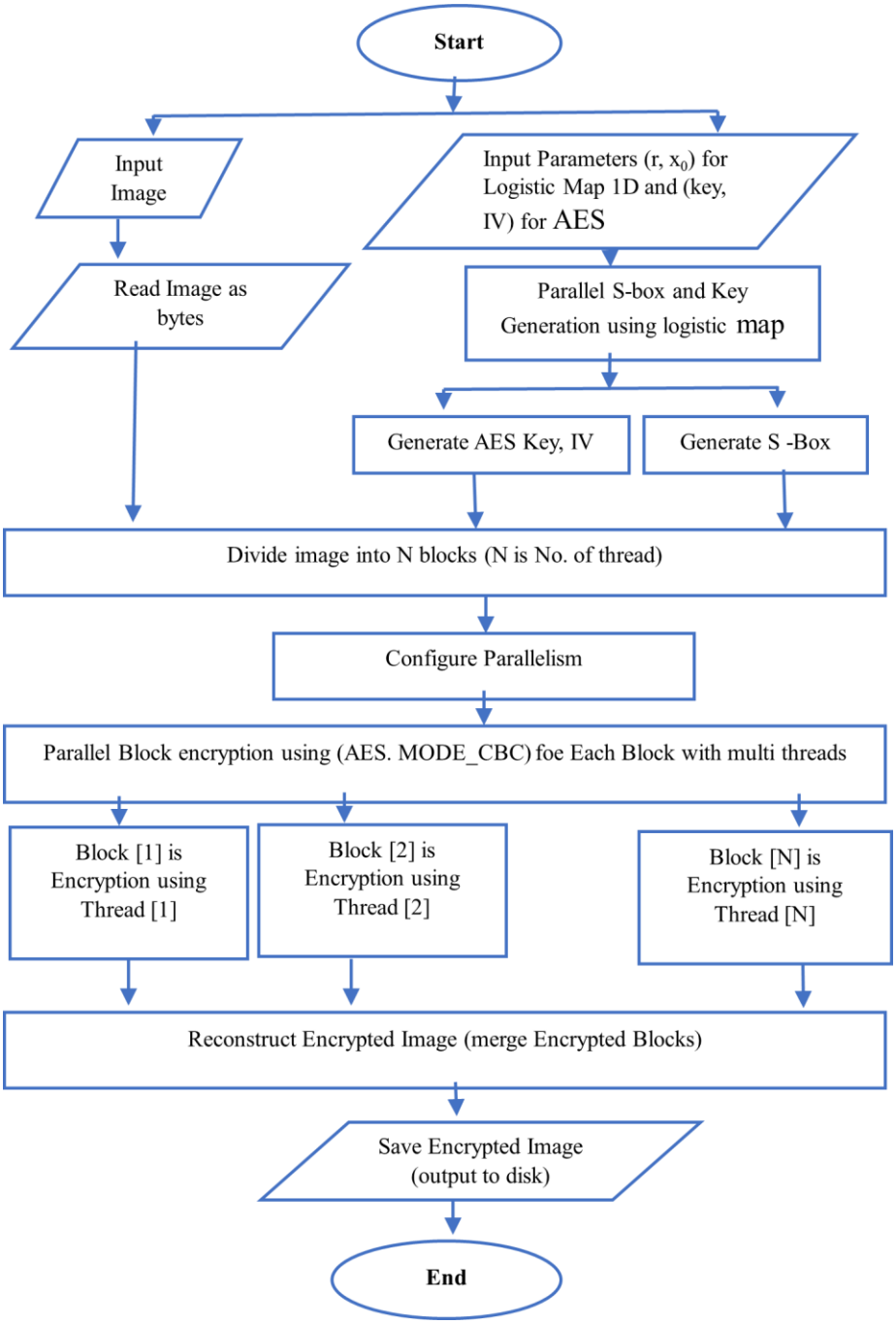


Figure 4: Main diagram of proposed method steps.

4. 3. 2 parallel AES decryption Parallel AES decryption is the reverse process of encryption.

4. 4 The Proposed Method's Benefits

- **Enhanced Security:** The encryption is robust to statistical and differential assaults because the logistic map adds randomization.
- **Parallel Efficiency:** AES encryption in parallel drastically cuts down on encryption and decryption time, particularly for large photos.
- **Scalability:** The technique can be readily expanded to include additional processing units for increased speed.

5 Results and Discussion

In this study, we tested the encryption performance of AES-CBC (Advanced Encryption Standard in Cipher Block Chaining mode) on several images, including Baboon, Bird, Peppers, Flower, Altun Kopri, and Panda. The encryption was carried out on a laptop with an Intel Core i5-6 processor, 8 GB of RAM, and a 64-bit Windows 10 operating system. Show in" Figure 5" below.

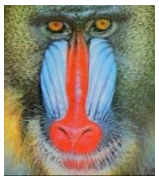
Image						
Name	Baboon	Peppers	Brid	Flower	Altun kopri-grey	Panda-grey
Size(bytes)	171 KB	110 KB	55.9 KB	1.61 MB	108 KB	50.7 KB
Resolution	512 x 512	512 x 512	620 x 413	1024 x1024	1080x 564	547 x 550

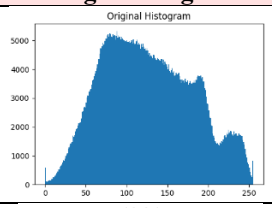
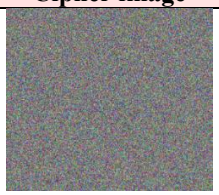
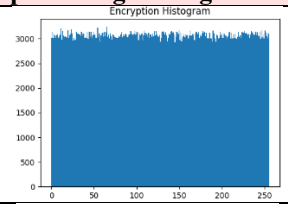
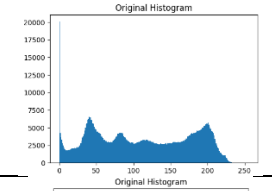
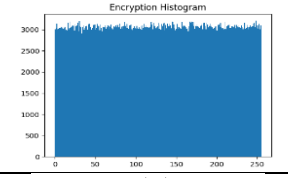
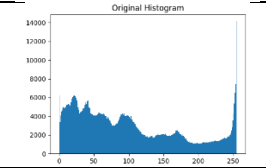
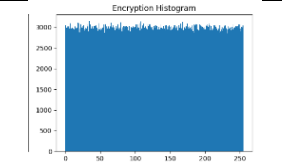
Figure 5: A collection of test images.

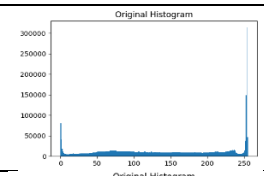
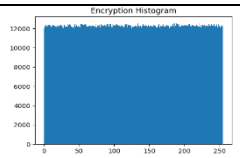
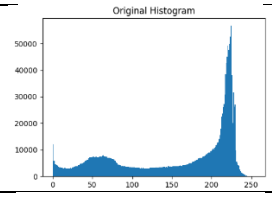
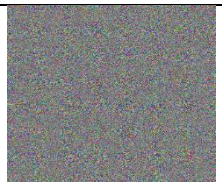
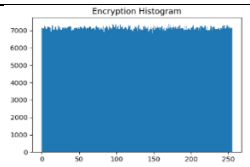
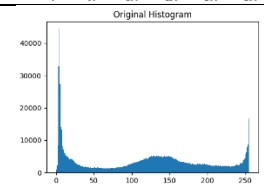
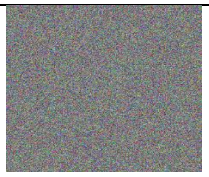
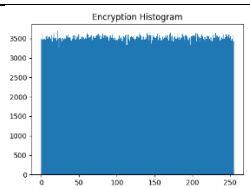
The performance of the current cryptosystem is assessed using the following standards:

5. 1 Histogram analysis

The histogram displays the image's pixel distribution. For the encrypted image, the graph needs to be consistent in order to withstand statistical attacks [17]. Five plain and encrypted photos' histograms are displayed in" Table 1". The distributions of the cipher images appear to be uniform, as can be shown. In both the horizontal and vertical axes, there is a high correlation between the adjacent pixels of the original image. For an ideal encryption scheme, the correlation coefficients of the pixels in the encrypted image can be sufficiently low to withstand statistical.

Table 1: displays histogram analysis; the number of pixels is represented by the y-axis in the histogram plot, while the gray-level values are represented by the x-axis.

Test Images	Test images histogram Plot	Cipher image	Cipher image histogram Plot
Baboon			
Peppers			
Bird			

Flower			
Altun kopri-grey			
Panda grey-scale			

5. 2 Avalanche criteria (AC)

The Avalanche Criterion (AC) is a crucial aspect in evaluating the performance of block ciphers. It looks at how even small changes in the input, such as altering a single bit, result in significant and unpredictable changes in the output. This property helps ensure the cipher's security by making the encrypted data look completely different from the original, making it harder to detect any patterns. The Avalanche Effect is vital for preserving data anonymity and protecting the cipher from cryptanalysis attacks. The output should be influenced by every bit of the input, and even a small change, like flipping a single bit, should result in a significant change (around 50%) in the output. [17]. The AC test results for the given cryptosystem are shown in "Table 2".

$$AC = \frac{\text{No.of Altering A Single Bit In Cipher Image}}{\text{No.of All Bits in Cipher image}} \times 100\ \% \qquad (2)$$

Table 2: Avalanche criteria (AC) test.

Images	AC With Key 16 Byte	AC With Key 24 Byte	AC With Key 32 Byte
Baboon	50.01 %	49.99 %	49.99 %
Peppers	50.00 %	49.97 %	50.00 %
Bird	50.01 %	50.02 %	50.01 %
Flower	49.99%	50.01 %	49.99 %
Altun kopri-grey scale	49.99 %	50.00 %	49.99 %
Panda- grey-scale	50.04 %	49.99 %	50.00 %

5. 3 NIST Randomness Test

To evaluate the robustness of the proposed encryption method, the study used the NIST test, a widely recognized standard for assessing the randomness and unpredictability of bit sequences. The test includes 16 different evaluations, with each test having a P-value. A P-value greater than 0.01 indicates that the sequence is considered random. The results, shown in "Table 3", revealed that the encryption technique successfully passed all 16 NIST tests for the Baboon image, demonstrating that the method produced a random and robust cipher text. [18].

Table 3: Baboon picture encryption is tested by NIST.

Type of Test	P-Value	Status	Type of Test	P-Value	Status
Frequency (Monobit) Test	0.19844	Random	**Maurer's "Universal Statistical" Test**	0.97787	Random
Frequency test within a block	0.67011	Random	**Linear Complexity Test**	0.14401	Random
Runs Test	0.72509	Random	**Serial Test:**	----------	----------
Longest run of ones in a block Test	0.65615	Random	**Approximate Entropy Test**	0.59510	Random

Binary matrix rank Test	0.41220	Random	**Cumulative Sums Test (Forward)**	0.27826	Random
Discrete Fourier Transform (Spectral) Test	0.24015	Random	**Random Excursions Test (-4)**	0.33229	Random
Non-overlapping Template Matching Test	0.78570	Random	**Random Excursions Variant Test (-9.0)**	0.51891	Random
Overlapping Template Matching Test	0.69912	Random			

5. 4 Entropy:

Entropy is a measure of the data's randomness in an image. An image with a high entropy value has more disordered data. One can measure the entropy using Equation (3), and the entropy value ought to be close to or equal to 8 [19]. Specifically, the following provides the equation:

$$\text{Entropy} = \sum_i P(S_x) \log_2 \left(\frac{1}{P(S_x)}\right) \quad (3)$$

In this case, the probability of pixel S_x (x=0 to 255) in a picture is denoted by $P(S_x)$. The entropy value, shown in "Table 4", is about 8. The likelihood of unintentional information leakage is reduced when the value is nearer 8.

Table 4: The results of Entropy.

Images	Entropy With Key 16 Byte	Entropy With Key 24 Byte	Entropy With Key 32 Byte
Baboon	7.9997	7.9997	7.9997
Peppers	7.9997	7.9997	7.9998
Bird	7.9997	7.9997	7.9997
Flower	7.9999	7.9999	7.9999
Altun kopri grey-scale	7.9998	7.9999	7.9998
Panda- grey-scale	7.9997	7.9998	7.9998

5. 5 Correlation analysis

The correlation between the original and encrypted photos is analyzed to determine how effective an encryption technique is. The recommended encryption method maintains low correlation values and outperforms previous methods. Using a mathematical formula, correlation values are determined and found to be around zero using Eq. (4), suggesting that there is no discernible correlation between neighboring boring pixels in any of the encrypted images[19]. The results demonstrate how well the suggested encryption approach maintains low correlation values, shown in "Table 5".

$$\text{Correlation} = \sum \left(\frac{(x - \mu x)(y - \mu y)}{\sigma_x \sigma_y}\right) \quad (4)$$

Table 5: The outcomes of the Correlation coefficients.

Images	correlation coefficients With Key 16 Byte	correlation coefficients With Key 24 Byte	correlation coefficients With Key 32 Byte
Baboon	0.00156	-0.00222	0.00084
Peppers	-0.00172	0.00151	-0.00018
Bird	0.00053	-0.00104	0.00036
Flower	0.00017	0.00093	-0.00016
Altun kopri grey-scale	0.00072	0.00013	0.00057
Panda- grey-scale	0.00023	-0.001341	0.00063

5. 6 PSNR and MSE analysis

According to "Table 6" for the Baboon, Peppers, and Bird photos, the PSNR and MSE values of the original and encrypted images are used to assess the encryption process's quality[20]. While a higher MSE value denotes a better encryption result, a lower PSNR number suggests higher encryption quality. Remembering that the decrypted picture should have an MSE of 0 and a PSNR of infinity is also crucial.

Table 6: PSNR and MSE values test.

Images	Encrypt With Key 16 Byte		Encrypt With Key 24 Byte		Encrypt With Key 32 Byte	
	MSE	*PSNR*	*MSE*	*PSNR*	*MSE*	*PSNR*
Baboon	8539.790	8.8163	8580.0253	8.7959	8569.3105	8.8013
Peppers	10069.8134	8.1005	10061.2216	8.1042	10069.2548	8.1008
Bird	11939.3554	7.3609	11926.8007	7.3655	11916.6376	7.3692
Flower	12659.6728	7.1065	12656.8310	7.1075	12666.9726	7.1040
Altun kopri grey-scale	11763.1347	7.4255	11765.6503	7.4246	11763.1484	7.4255
Panda- grey-scale	12321.804	7.2240	12342.5546	7.21675	12312.8046	7.2272

5. 7 Differential Attack

NPCR and UACI, which were computed using Equations (5) and (6) for M×H image size, were employed to evaluate the vulnerability of the proposed image encryption technique against differential attacks. According to the results, which are displayed in "Table 7", the proposed technique is very sensitive to a one-pixel change and outperforms previous studies [19].

$$NPCR = \left[\frac{1}{M.H}\sum_{x,y} D(x,y)\right].100\% \quad (5)$$

$$UACI = \frac{1}{M*H}\left[\sum_{x,y}\frac{|C_1(x,y) - C_2(x,y)|}{255}\right] * 100\ \% \quad (6)$$

$$D(x,y) = \begin{cases} 1, & C_1(x,y) \neq C_2(x,y) \\ 0, & C_1(x,y) = C_2(x,y) \end{cases} \quad (7)$$

Variable C1 represents the encryption of the original image, while variable C2 represents the encryption of the original image after a single pixel change.

Table 7: Differential Attack NPCR and UAC values test.

Images	With key 16 byte		With key 24 byte		With key 32 byte	
	NPCR	UACI	NPCR	UACI	NPCR	UACI
Baboon	99.6103	29.8016	99.6126	29.8790	99.6014	29.8656
Peppers	99.6128	32.1594	99.6173	32.1332	99.5997	32.1586
Bird	99.5881	35.0338	99.6172	35.0107	99.6159	34.9865
Flower	99.6143	36.1226	99.6116	36.1169	99.6091	36.1322
Altun kopri -grey-scale	99.6064	34.7467	99.6083	34.7433	99.6096	34.7524
Panda- grey-scale	99.6061	35.5995	99.6145	35.6466	99.6074	35.6041

5. 8 Analysis based on Execution Time

The execution time analysis of a parallel AES (Advanced Encryption Standard) implementation with a block size of 16 bytes and a combination with a logistic map for image encryption and decryption[19,20]. Shown in Table 8.

Table 8: Encryption and Decryption Time Comparison(S).

Images	With key 16 byte		With key 24 byte		With key 32 byte	
	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt
Baboon	0.01562	0.0039	0.0030	0.0019	0.0039	0.0019
Peppers	0.00289	0.0020	0.0029	0.0009	0.004	0.004
Bird	0.01562	0.0010	0.0039	0.0009	0.0029	0.0019
Flower	0.01568	0.0070	0.0089	0.0069	0.0089	0.0079
Altun kopri grey-scale	0.01562	0.0039	0.0059	0.0039	0.0060	0.0039
Panda- grey-scale	0.00399	0.0030	0.0039	0.0030	0.0039	0.0019

5. 9 Parallel metrics and analysis

The analysis of a hybrid system involves parallel AES encryption/decryption using a logistic map. Below is an explanation of each metric[20], shown in "Table 9", "Table 10", "Table 11".

2. 9. 1 Throughput (MB/s): A higher throughput means that data is processed by the system more quickly. Hardware acceleration or improved parallelization can provide improvements.

$$\text{Throughput (MB/s)} = \frac{\text{Execution Time}}{\text{Total Data Size}} \quad 8$$

2. 9. 2 Memory Usage (MB): Depending on parallelization (e.g., the number of threads employed), memory usage usually rises with data size.

2. 9. 3 Speed-Up (S): Shows how much performance is enhanced by parallel execution. The workload and hardware determine the optimal speedup[9,20].

$$\text{Speed} - \text{Up (S)} = \frac{\text{Execution Time (Sequential)}}{\text{Execution Time (Parallel)}} \quad 9$$

2. 9. 4 Efficiency (%): Because of parallel overhead (such as synchronization charges), this falls as the number of cores rise[20].

$$\text{Efficiency (\%)} = \frac{\text{Speed} - \text{Up (S)}}{\text{Number of Cores}} \times 100 \quad 10$$

2. 9. 5 Data Processing Factor (bytes/sec): Aids in comparing the processing speeds of raw data across various configurations or systems[14,20].

$$\text{DPF (bytes/sec)} = \frac{\text{Total Data Size (bytes)}}{\text{Execution Time (s)}} \quad 11$$

Table 9: Parallel metrics and analysis (with key 16 Byte).

Images	Throughput (MB/s)	Memory Usage (MB)	Speed-Up (S)	Efficiency	Data Processing Factor (DPF) (bytes/sec)
Baboon	47.9921	105.59	3.9093	74.42 %	40072914.93
Peppers	259.4467	105.67	1.4446	30.78%	160766253.64
Bird	250.1822	107.06	15.5901	93.59 %	46197301.29
Flower	191.2800	147.91	3.9064	55.35%	138655779.25
Altun kopri grey-scale	249.73535	124.12	1.7480	74.40%	93122864.27
Panda- grey-scale	215.2668	107.71	1.3042	23.33%	127762774.83

Table 10: Parallel metrics and analysis (with key 24 Byte).

Images	Throughput (MB/s)	Memory Usage (MB)	Speed-Up (S)	Efficiency	Data Processing Factor (DPF) (bytes/sec)
Baboon	249.9636	105.75	1.5007	33.37 %	157293500.18
Peppers	250.1822	105.82	3.0023	66.69 %	196790478.00
Bird	183.1972	105.35	4.0002	75.00 %	153678850.44
Flower	333.3856	146.50	1.2860	22.25 %	196664281.45
Altun kopri grey-scale	290.5208	123.95	1.5002	33.34 %	182790070.98
Panda- grey-scale	215.3824	107.72	1.3310	24.87 %	128959666.78

Table 11: Parallel metrics and analysis (with key 32 Byte).

Images	Throughput (MB/s)	Memory Usage (MB)	Speed-Up (S)	Efficiency	Data Processing Factor (DPF) (bytes/sec)
Baboon	187.5509	105.71	1.9998	50.00 %	131105007.64
Peppers	187.3722	106.02	1.0002	0.03 %	98251630.54
Bird	244.3941	105.37	1.5119	33.86 %	154245333.83
Flower	333.3415	147.94	1.1306	11.56 %	185486435.86
Altun kopri grey-scale	290.4400	123.83	1.5009	33.38 %	182776994.00
Panda- grey-scale	215.1898	107.46	2.0020	50.05 %	150479429.91

5. 10 Comparison

In "Table 12", Regarding NPCR, entropy, and correlation, the suggested approach performs marginally better, suggesting improved random and encryption strength. It is challenging to do direct comparisons because the PSNR data are only provided for the suggested approach. Previous research's pixel intensity varies more, as indicated by their higher UACI values. The suggested approach has much faster encryption and decryption speeds, indicating more efficiency.

Table 12: Comparison with previous studies.

Measurements	Proposed		[11]		[14]	
Image	**Baboon**	**Peppers**	**Baboon**	**Peppers**	**Baboon**	**Peppers**
Size image	**512 x 512**	**512 x 512**	**512 x 512**	**512 x 512**	**512 x 512**	**512 x 512**
PSNR	8.8163	8.1005	-	-	-	-
NPCR	99.6103	99.6128	99.604	99.594	99.639	99.627
UACI	29.8016	32.1594	33.5662	33.5772	33.472	33.467
Entropy	7.9997	7.9997	7.9978	7.9966	7.99932	7.99936
Correlation	0.00156	-0.00172	-0.002810	0.00304	8.8929	-0.0021
Encryption speed (s)	0.01562	0.00289	-	-	0.06997	0.07201
Decryption speed (s)	0.0039	0.0020	-	-	0.09495	0.00997

6 Conclusion

Image encryption is a crucial aspect of modern communication systems, and a robust and efficient approach has been proposed by combining parallel AES-CBC mode with multi-threading. This method uses a chaotic logistic map for dynamic S-box and key generation, enhances the security of the AES algorithm by addressing vulnerabilities associated with static components. The integration of CBC mode further optimizes the encryption process, reducing encryption time and ensuring real-time applications. The proposed method achieves high key sensitivity, low correlation coefficients, and resistance to attacks, making it a promising solution for safeguarding sensitive visual data.

References

[1] Rashid, A. A. & Hussein, K. A. A Lightweight Image Encryption Algorithm Based on Elliptic Curves and a 5D Logistic Map. *Iraqi J. Sci.* **64**, 5985–6000 (2023).

[2] Hussein, K. A. & Taha, M. D. A new hybrid lightweight algorithm (RC6-Salsa20) to image encryption. in *AIP Conference Proceedings* **3282** (AIP Publishing, 2025).

[3] Arab, A., Rostami, M. J. & Ghavami, B. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **75**, 6663–6682 (2019).

[4] Rahman, Z., Yi, X., Billah, M., Sumi, M. & Anwar, A. Enhancing AES Using Chaos and Logistic Map-Based Key Generation Technique for Securing IoT-Based Smart Home. *Electron.* **11**, 1–15 (2022).

[5] Abdullah, A. M. Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. *Cryptogr. Netw. Secur.* **16**, 11 (2017).

[6] Zeghid, M., Machhout, M., Khriji, L., Baganne, A. & Tourki, R. A Modified AES Based Algorithm for Image Encryption. *Int. J. Comput. Sci. Eng.* **1**, 70 (2007).

[7] Fadhil, M. S., Farhan, A. K. & Fadhil, M. N. A lightweight aes algorithm implementation for secure iot environment. *Iraqi Journal of Science* vol. 62 2759–2770 (2021).

[8] Muhammed, N. T. *et al.* Optimizing Time Consumption for Smartphone-based Distributed Parallel Processing System. *Passer J. Basic Appl. Sci.* **6**, 254–265 (2024).

[9] Le, D., Chang, J., Gou, X., Zhang, A. & Lu, C. Parallel AES algorithm for fast data encryption on GPU. *ICCET 2010 - 2010 Int. Conf. Comput. Eng. Technol. Proc.* **6**, 1–6 (2010).

[10] Rahman, Z., Yi, X., Khalil, I. & Sumi, M. Chaos and Logistic Map Based Key Generation Technique for AES-Driven IoT Security. *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST* **402 LNICST**, 177–193 (2021).

[11] Salman, R. S., Farhan, A. K. & Shakir, A. Creation of S-Box based One-Dimensional Chaotic Logistic Map: Colour Image Encryption Approach. *Int. J. Intell. Eng. Syst* **15**, 2022 (2022).

[12] Fadhil, M. S., Farhan, A. K. & Fadhil, M. N. Designing Substitution Box Based on the 1D Logistic Map Chaotic System. *IOP Conf. Ser. Mater. Sci. Eng.* **1076**, 012041 (2021).

[13] Arpay, M. N. & Talirongan, H. B. Enhancing Advanced Encryption Standard (AES) Security through Logistic Map Integration. **1**, (2019).

[14] Abdulkadhim, A. A. A. & Lakizadeh, A. A Parallel Image Encryption Method Based on Hybrid Chaotic Technique. *Int. J. Intell. Eng. Syst.* **17**, 836–847 (2024).

[15] Assafli, H. T., Hashim, I. A. & Naser, A. A. Advanced Encryption Standard (AES) acceleration and analysis using graphical processing unit (GPU). *Appl. Nanosci.* **13**, 1245–1250 (2023).

[16] Nabil, M., Khalaf, A. A. M. & Hassan, S. M. Design and implementation of pipelined and parallel AES encryption systems using FPGA. *Indones. J. Electr. Eng. Comput. Sci.* **20**, 287–299 (2020).

[17] Abdalrahman, A. O., Jabbar, K. K., Karim, A. B. & Abdulhammed, O. Y. Secure Communication of the Integrated IoT and Cloud Computing. *Passer J. Basic Appl. Sci.* **4**, 40–50 (2022).

[18] Hoomod, H. K., Naif, J. R. & Ahmed, I. S. A new intelligent hybrid encryption algorithm for IoT data based on modified PRESENT-Speck and novel 5D chaotic system. *Period. Eng. Nat. Sci.* **8**, 2333–2345 (2020).

[19] Taha, M. & Hussein, K. A. Design Sequential Hybrid Lightweight Algorithms For Encryption Data Using RC6 Algorithm. *Mustansiriyah J. Pure Appl. Sci.* **3**, 1–11 (2025).

[20] Taha, M. D. & Hussein, K. A. A New Approach to Design KM1 Encryption Algorithm for Alarm in IoT. in *National Conference on New Trends in Information and Communications Technology Applications* 395–410 (Springer, 2023).