Mesopotamian journal of Cybersecurity
Vol.5,No.2, **pp**. 436–452
DOI: https://doi.org/10.58496/MJCS/2025/027; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Color Image Encryption Based on a New Symmetric Lightweight Algorithm

Ala'a Talib Khudhair[1,*], Abeer Tariq Maolood[1], Ekhlas Khalaf Gbashi[1]

[1] *Computer Science Department, University of Technology, Al-Sina'a St., Al-Wehda District, Baghdad 10066, Iraq*

ARTICLE INFO

Article History

Received 12 Dec 2024
Revised 12 Mar 2025
Accepted 24 Mar 2025
Published 06 Jun 2025

Keywords

Color image
Confusion
Lightweight algorithm
F-Function
Gauss map

ABSTRACT

Many lightweight algorithms, such as the tiny lightweight algorithm, have significant weaknesses, mainly due to the lack of substitution boxes, effective confusion mechanisms, or both. In today's world, enhancing encryption and secure transmission has become increasingly vital. This paper presents a newly developed lightweight algorithm for color image encryption that is based on a new symmetric block cipher structure. The method starts by transforming each pixel channel value into a 24-bit binary number. A new F-function is introduced in this block cipher to improve diffusion and confusion. Additionally, a 3D Hindmarsh-Rose model is used to generate a dynamic 6-bit S-Box in an octal format (8×8). A new approach, which is based on the Gauss map, is proposed for generating shift values, which further enhances confusion in the block cipher alongside additive and XOR operations. Python simulation experiments were conducted to analyse the security of the encryption. Tests were performed on the Lena image with a resolution of 512×512 pixels, yielding information entropy values of 7.9992 for red, 7.9990 for green, and 7.9992 for blue. The correlation coefficients were minimal, with values of red (horizontal: -0.0027, vertical: -0.0011, diagonal: 0.0019), green (horizontal: -0.0027, vertical: -0.0015, diagonal: 0.0020), and blue (horizontal: 0.0023, vertical: 0.0031, diagonal: 0.0025). Additionally, differential attack tests, including the number of pixel change rates (NPCRs) and unified average change intensities (UACIs), yielded values of 99.6048, 99.6090, and 99.6014 for the NPCRs and 33.3680, 33.4909, and 33.4099 for the UACIs across the red, green, and blue channels, respectively. The results demonstrate that the proposed algorithm provides strong encryption performance and high resistance to differential attacks.

1. INTRODUCTION

Securing images has become a crucial topic in information security, primarily aiming to ensure the confidentiality and integrity of visual data. With the prevalent use of digital images, robust encryption techniques are vital to protect against unauthorized access, modification, and interception during transmission and storage. Over the past few years, numerous image encryption techniques have been developed to serve various purposes. These techniques are symmetric, asymmetric, and chaotic and are applicable to both grayscale and RGB images. Grayscale images, which are commonly used in fields such as medical imaging, often contain sensitive data that must be securely protected [1]. Encrypting a grayscale image typically involves applying a mathematical transformation with a secret key, rendering the image unreadable. Only users with the proper key are authorized to decrypt the image and return it to its original form. Likewise, RGB images also need safeguarding because of the sensitive content they may contain. The encryption process for RGB images follows the same approach, using a secret key to transform the image into an unreadable format, and decryption is possible only for those with the correct key [2].

This paper introduces a newly developed lightweight algorithm for color image encryption that employs a new symmetric block cipher structure. This structure incorporates a new F-function to enhance diffusion and confusion. Additionally, a 3D Hindmarsh-Rose model is utilized to dynamically generate a 6-bit S-Box (8×8) in an octal format. To further strengthen confusion within the block cipher, shift values are generated via a new approach that is based on the Gauss map in combination with additive and XOR operations.

*Corresponding author. Email: cs.22.20@grad.uotechnology.edu.iq

The rest of this paper is structured as follows: Section 2 presents a lateral review, Section 3 provides a theoretical background, and Section 4 presents the proposed methodology. Section 5 presents the security analysis of the proposed algorithm. Finally, Section 6 provides conclusions and discusses potential avenues for future research.

2. LITERATURE REVIEW

With the swift advancement of information technology, safeguarding the security of information transmission has become increasingly important. Digital images, which are commonly used for transmitting information, play crucial roles in various sectors, such as healthcare, military, industry, and everyday activities. Recently, chaotic systems have emerged as a significant method for securely encrypting images [3]. These systems are highly sensitive to input variations, enabling the generation of highly secure cipher images by applying the information produced by one or more chaotic systems to plain images [4]. For example, [5] proposed the use of an evolutionary codebook and chaotic dystems to encrypt color images. In [6], an algorithm was introduced to encrypt color images via a bit-plane and chen chaotic system. In [7], a color image encryption algorithm based on hybrid two-dimensional hyperchaos and genetic recombination was proposed. In [8], a new image encryption algorithm based on a four-wing chaotic system was proposed. In [9], a new S-Box generation method based on a hybrid two-dimensional chaotic map for color image encryption was proposed. In [10], the generation of dynamic substitution boxes via the HSM chaos system was proposed for application in color image encrypting. However, all these methods exhibit low levels of entropy, correlation analysis, NPCRs, and UACIs, indicating that the resulting ciphertext lacks sufficient randomness. In [11], an image encryption algorithm using a 5-D hyperchaotic system and a DNA sequence was proposed. This work achieves good results in terms of entropy, correlation analysis, NPCR, and UACI. However, it uses a 5-D hyperchaotic system along with DNA, which may increase the execution time when running for many periods. As a result, it may not be suitable for lightweight algorithms.

TABLE I. COMPARISON OF PREVIOUS RELATED WORKS

Ref.	Methodology	Positive aspects	Negative aspects
[5]	Color image encryption based on an evolutionary codebook and chaotic systems	The encryption achieves entropy values near the theoretical maximum 8, ensuring high randomness. Additionally, it shows good NPCR and UACI results	The encryption process relies on a continuously evolving codebook. For longer subsequences, a codebook with 2^l entries must be maintained, leading to exponential growth in size. This poses challenges in storage and computational efficiency, making it necessary to keep l as a small integer
[6]	Color image encryption algorithm based on Bit-Plane and Chen chaotic system	The proposed algorithm achieved good entropy, NPCR, and UACI results, demonstrating its resilience against statistical and differential attacks	The proposed algorithm is more intricate than traditional lightweight encryption techniques like the Tiny Encryption Algorithm (TEA) or Lightweight Encryption Algorithm (LEA), making it less suitable for applications requiring minimal computational overhead
[7]	A novel color image encryption algorithm based on hybrid two-dimensional hyperchaos and genetic recombination	The proposed algorithm achieved good NPCR, and UACI results,	The proposed algorithm faces challenges when encrypting high-resolution images, as the key length needed increases in proportion to the image size. As a result, encrypting large, high-precision images demands longer keys, which in turn places greater demands on computer hardware.
[9]	New S-Box generation based on hybrid two-dimensional chaotic map for color image encryption	The proposed method utilizes a hybrid two-dimensional chaotic map for color image encryption, thereby strengthening security through enhanced nonlinearity	The study lacks a detailed analysis of the S-Box construction, specifically in terms of evaluating the Linear Branch Number (LBN) and Differential Branch Number (DBN) to demonstrate its strength against linear and differential attacks. Additionally, the proposed algorithm does not achieve satisfactory results in entropy, NPCR, and UACI metrics
[10]	Generation of dynamic substitution boxes using HSM chaos system for application in color images encrypting	The proposed method presents a new chaotic map that combines the Hénon and Sine maps for S-Box construction	The color image encryption approach lacks a diffusion-enhancing step in the encryption process, such as the Initial Permutation (IP) in the DES algorithm. Additionally, it does not incorporate a function that simultaneously achieves both diffusion and confusion, similar to the F function in the Blowfish or FEAL algorithms
[11]	A novel color image encryption algorithm based on 5-D hyperchaotic system and DNA sequence	The proposed method achieved good results in entropy, correlation analysis, NPCR, and UACI	The proposed method, which employs a 5-D hyperchaotic system, requires more execution time, making it less suitable for lightweight encryption

In addition, many lightweight algorithms face significant weaknesses, primarily due to their reliance on static substitution boxes (S-boxes) or the absence of effective confusion mechanisms. Algorithms that lack an S-Box or fail to incorporate a

suitable confusion function tend to exhibit weaker security properties, as highlighted in Table II. This underscores the importance of employing robust cryptographic structures to ensure secure and efficient performance.

TABLE II. FIXED S-BOX, LACK AN S-BOX OR LACK A FUNCTION IN LIGHTWEIGHT ALGORITHMS

Algorithm	Static S–Box	lack an S-Box	lack a function
Advance Encryption Standard	✓		✓
Blowfish	✓		
CAST	✓		✓
Scalable Encryption Algorithm (SEA)	✓		✓
Data Encryption Standard (DES)	✓		
PRESENT	✓		✓
GOST	✓		✓
HIGHT		✓	✓
International DES (IDES)		✓	✓
SAFER		✓	✓
RC5		✓	✓
MISTY		✓	✓
KATAN		✓	✓
SPECK		✓	✓
SIMON		✓	✓
SIMECK		✓	✓
SPARX		✓	✓
XSX		✓	✓
CHAM		✓	✓
JAC Jo		✓	✓
BRIGHT		✓	✓

3. THEORETICAL BACKGROUND

3.1 The 1D Gauss map

is formally described by Eq. (1), which defines the 1D Gauss map iterator [12]:

$$x_{n+1} = \exp(-\alpha \times x_n^2) + \beta \quad (1)$$

Two parameters are involved: α and β. The range for α is greater than 0, whereas β ranges from -1 to +1.

3.2 The 3D Hindmarsh-Rose model

is formally described by Eq. (2), which defines the 3D Hindmarsh-Rose iterator [13].

$$\begin{cases} x_{n+1} = y - a * x^3 + b * x^2 - z + I \\ y_{n+1} = c - d * x^2 - y \\ z_{n+1} = r\,[\,s\,(x - xr) - z\,] \end{cases} \quad (2)$$

The model consists of eight parameters: a, b, c, d, r, s, xr, and I. Common practice involves fixing certain parameters while using others as control variables. The parameter I is usually treated as a control variable, and a, b, c, d, or r are also commonly used as control parameters in existing studies. Typically, the values s = 4 and xr = -8/5 are set as constants. When fixed, the values of a, b, c, and d are a = 1, b = 3, c = 1, and d = 5. The parameter r typically varies between 0.001 and 0.003, whereas I ranges from -10–10 [13].

3.3 Concepts of Confusion and Diffusion in Cryptography

Successful block cipher designs often incorporate the principles of confusion and diffusion.

- ✓ Confusion obscures the relationship between plaintext and ciphertext, making it difficult to deduce the original plaintext [33]. This is achieved through substitution, where a binary word is replaced by another binary word, resulting in a ciphertext that appears meaningless [14].
- ✓ Diffusion distributes the plaintext statistics throughout the ciphertext, ensuring that changes in the plaintext are reflected across the entire ciphertext [34]. This is implemented via permutation, where the bits of a binary word are reordered [15].

4. THE PROPOSED METHODOLOGY

This paper introduces a newly developed lightweight algorithm designed for symmetric encryption. The proposed algorithm is a 9-byte (72-bit) block cipher requiring a key (108 bits): subkey1 (36 bits) and subkey2 (72 bits) to encrypt the color image algorithm that contributes to causing diffusion and confusion in the block cipher. For the encryption of a color image, this method begins by converting the value of each pixel channel into a 24-bit binary number; the encryption process is illustrated in Figure 1 and detailed in Algorithm 1. Decryption follows the same structure but is applied in the opposite sequence.

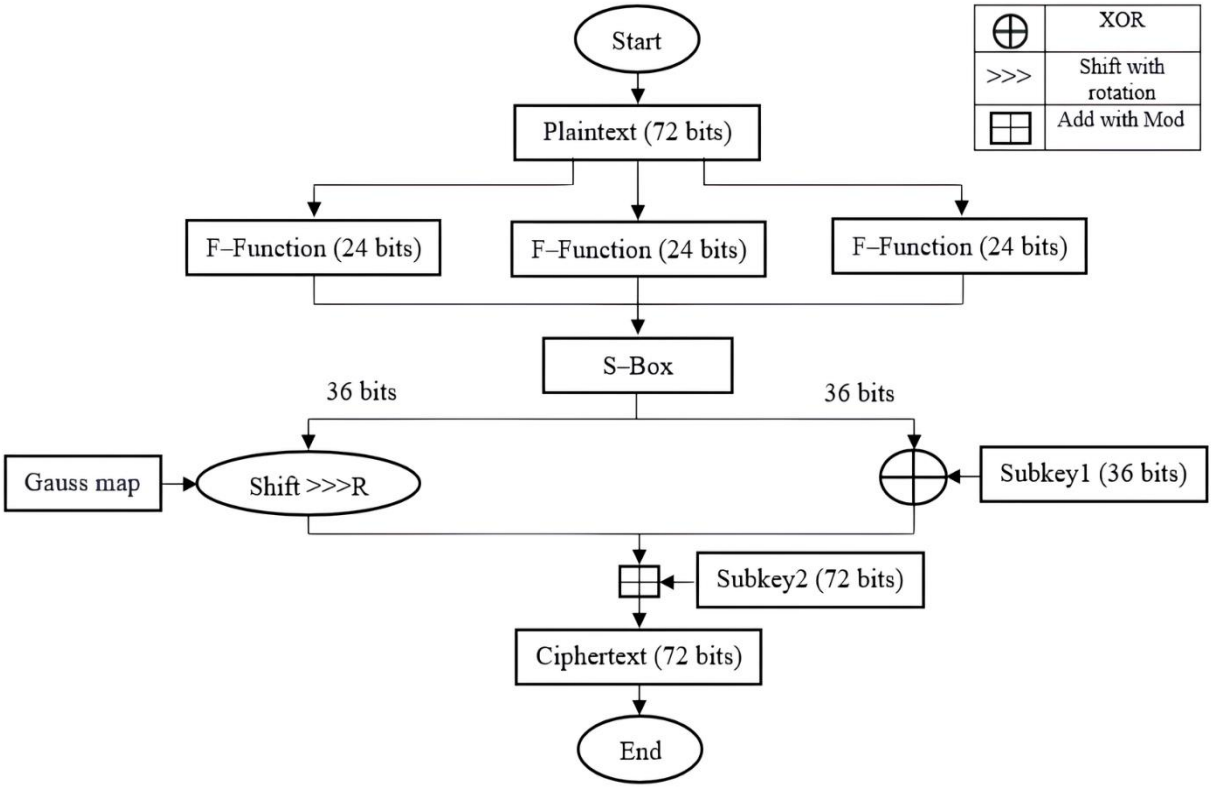


Fig. 1. One round of the proposed structure of the new lightweight encryption algorithm

Algorithm 1 Proposed new lightweight algorithm for color image encryption

Input: Plaintext (72 bits), key (108 bits) {subkey1 (36 bits), and subkey2 (72 bits)}.
Output: Ciphertext (72 bits).
Begin

1. Read the plaintext block (72 bits).
2. For round = 1 to 4
3. Split the plaintext into three parts, each consisting of 24 bits, and input them into the F-Function.
4. Apply the S-Box to the output from step 3.
5. Split the output from step 4 into two halves (Right and Left), each 36 bits in length. XOR the right halves with subkey1, while the left halves are shifted using the Gauss map.
6. Concatenate the results from step 5 and then additive with subkey2 to produce the ciphertext.
7. The result of step 6 be new plaintext.
8. Using new key of length 108 bits.
9. Next for

End

4.1 F–Function

A new F–Function is developed in this algorithm that contributes to causing diffusion and confusion in the block cipher, as shown in Figure 2. The input to this function is 24 bits; the data block is broken up into 6-bit chunks, and then the chunks are shifted and XORed to produce the output. The same function is used for decryption but in reverse order.

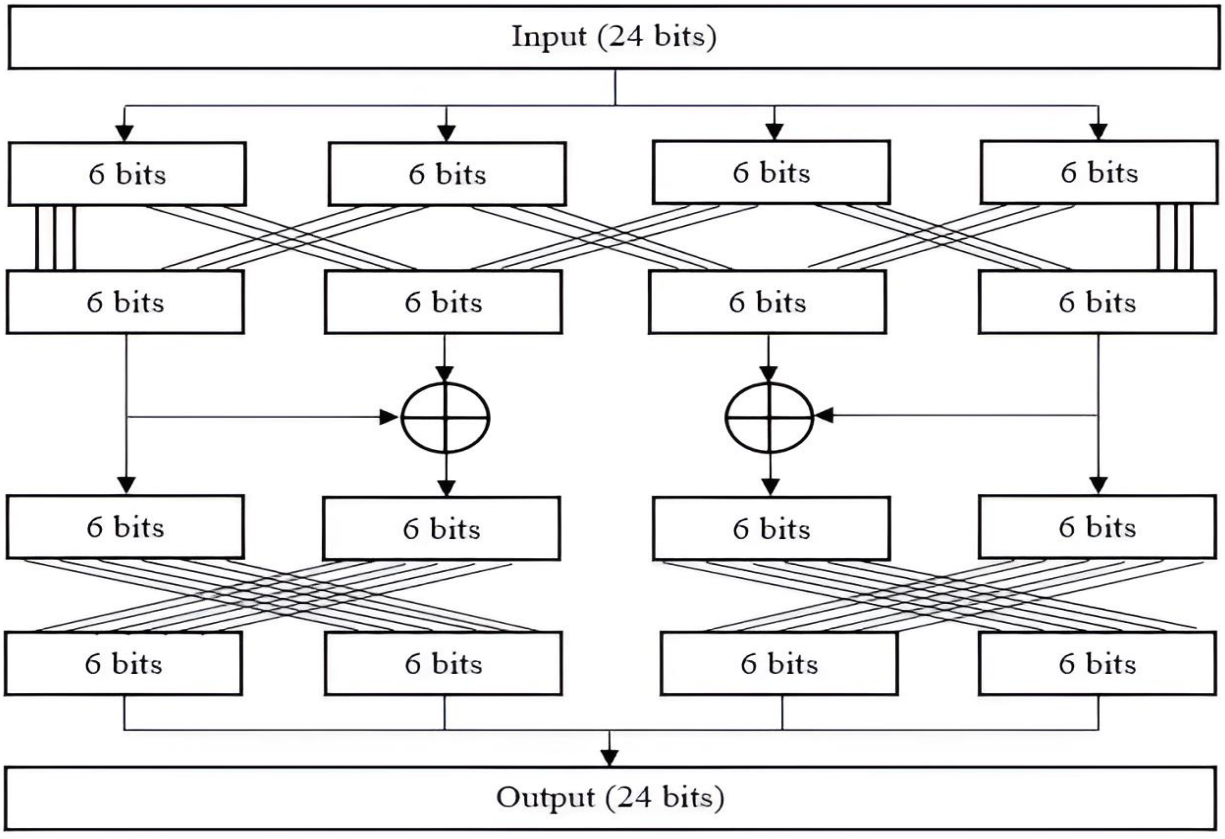


Fig. 2. Inner structure of the F-Function

4.2 Generate Shift Values Based on the 1D Gauss Map

In this algorithm, a new method was proposed for generating shift values on the basis of a Gauss map that is responsible for enhancing confusion in a block cipher, as shown in Algorithm 2 and Figure 3. In this manner, the shift values will be variable, ensuring that any manipulation of the initial values of the 1D Gauss map results in the creation of new shift values. This variability enhances the algorithm's resistance to attacks. The 1D Gauss map was chosen for its simplicity, which makes it suitable for lightweight algorithms.

Algorithm 2 Generating shift values using the Gauss map

Input: Initial parameters for the 1D Gauss map (α, β, and X_0)
Output: Shift values (buffer).
Begin

1. Read and initialize the given starting conditions.
2. For i = 1 to n, where n is a variable defined by the user.
   2.1: Calculate Xi using the 1D Gauss map formula (Eq. 1).
   2.2: Move to the next iteration (i).
3. Generates shift values (buffer) depending on X_i:
   3.1: For j = 1 to 6
   3.2: Randomly select a value of Xi, remove the sign, and take the first 14 digits after the decimal point.
   3.3: Randomly choose a position within the 14 digits, read its value, apply mod 6, and store the result in the buffer.
   3.4: Move to the next iteration (j).

End

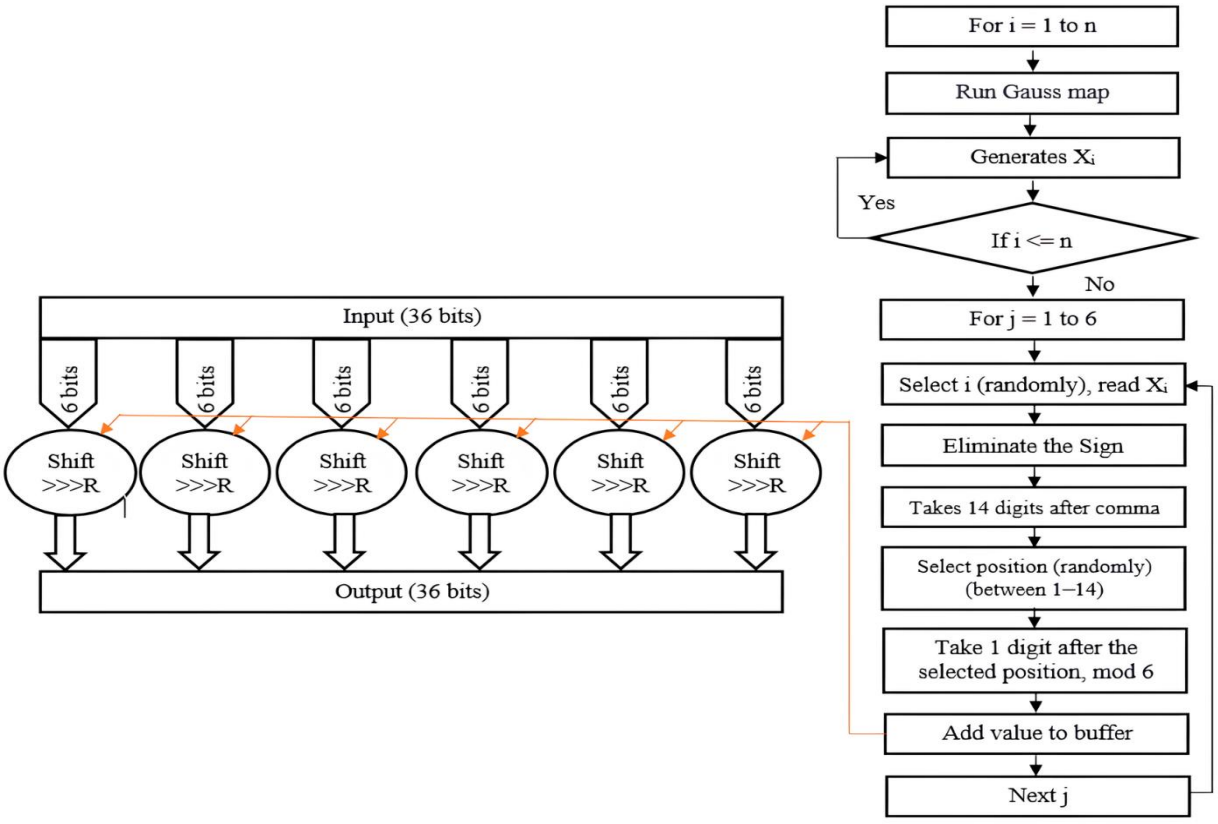


Fig. 3. Shift based on the Gauss map

➢ The following is a full example illustrating the generation of shift values via the 1D Gauss map:

If both the sender and recipient mutually consent to generate 500 random numbers via the 1D Gauss map with $\alpha = 4.9$, $\beta = -0.58$, and $X_0 = 0.2$.

When i = 1 $X_1 = e^{-4.9 \times 0.2^2} + (-0.58) = 0.24201223467818656$

The remaining results are derived via the same method, as demonstrated in Table III.

TABLE III. GENERATE Xi VIA THE 1D GAUSS MAP

Number of i	X_i
1	0.24201223467818656
2	0.17051727026368591
3	0.287210896394552
4	0.087509614452014661
5	0.38317142669323612
6	-0.092965188506095142
…	…
500	-0.0068751800909075955

➢ If both the sender and receiver consent to randomly select values for i and position, Table IV demonstrates how shift values (buffers) are generated

TABLE IV. SHIFT VALUES (BUFFER) ARE GENERATED VIA THE 1D GAUSS MAP

j	Selected i	Value of (X_i)	Remove the sign, and take the first 14 digits after the decimal point.	Selected Position	Shift values (buffer)
1	3	0.287210896394552	28721089639455	4	2 mod 6 = 2
2	150	-0.158276607978375	15827660797837	7	6 mod 6 = 0
3	299	0.337049547606399	33704954760639	9	7 mod 6 = 1
4	305	0.405335678179559	40533567817955	10	1 mod 6 = 1
5	400	0.0549079110815289	05490791108152	12	1 mod 6 = 1
6	499	0.337049548285601	33704954828560	2	3 mod 6 = 3

The shift values in the buffer are “201113”

- Here is an example of how to perform bit shifting on the basis of the shift values generated by the 1D Gauss map.

Input (36–bit)	101001100101110010100100101001100101					
6–bit before shift	101001	100101	110010	100100	101001	100101
Shift values	2	0	1	1	1	3
6–bit after shift	011010	100101	011001	010010	110100	101100
Output (36–bit)	011010100101011001010010110100101100					

4.3 S–Box

In this work, a 3D Hindmarsh Rose model was employed to create a dynamic 6-bit S-Box in octal format (8×8); the generation process was carried out in two stages. The initial stage involves number initialization, as described in Algorithm 3. The subsequent stage focuses on constructing the S-Box, as detailed in Algorithm 4. The 3D Hindmarsh Rose model was chosen for its simplicity, which makes it suitable for lightweight algorithms.

Algorithm 3 Number initialization phase

Input: Initial parameters for the 3D Hindmarsh-Rose model (a, b, c, d, r, s, xr, I, X_0, Y_0 and Z_0).
Output: sequence 1, sequence 2, and sequence 3
Begin

1. Read and initialize the given starting conditions.
2. Number initialization phase:
   2.1: For i = 1 to n, where n is a variable defined by the user.
   2.2: Generates the value X_i and extract 13 digits from the result to form sequence 1.
   2.3: Use the newly generated Sequence 1 to compute Yi, then extract 13 digits from this result to create Sequence 2.
   2.4: Similarly, compute Zi using Sequence 1 and extract 13 digits from the result to form Sequence 3.
   2.5: Increment i and repeat the process.
3. Remove any signs and decimal points from the generated Sequence 1, Sequence 2, and Sequence

End

Algorithm 4 S-Box construction phase

Input: Sequence 1, sequence 2, and sequence 3
Output: An (8×8) S–Box.
Begin

1. For i = 1 to n
2. Randomly select a round (i) and retrieve its value.
3. Choose randomly from Sequence 1, Sequence 2, or Sequence 3 and read the corresponding value.
4. Randomly pick a position, ensuring it does not exceed 10.
5. Extract two digits starting from the selected position, then apply mod 64 to the result.
6. Convert the mod result into a 6-bit binary format.
7. Transform the 6-bit binary value into its octal equivalent.
8. If the octal value from Step 6 is not already present in the S-Box, add it.
9. Increment i and repeat the process until the S-Box (8×8) is complete.

End

4.4 Key Generation

This paper introduces a new approach for generating long symmetric keys tailored for lightweight cryptographic algorithms that employs a pretrained visual geometry group 16 (VGG16) symmetric key, as shown in Figure 4.

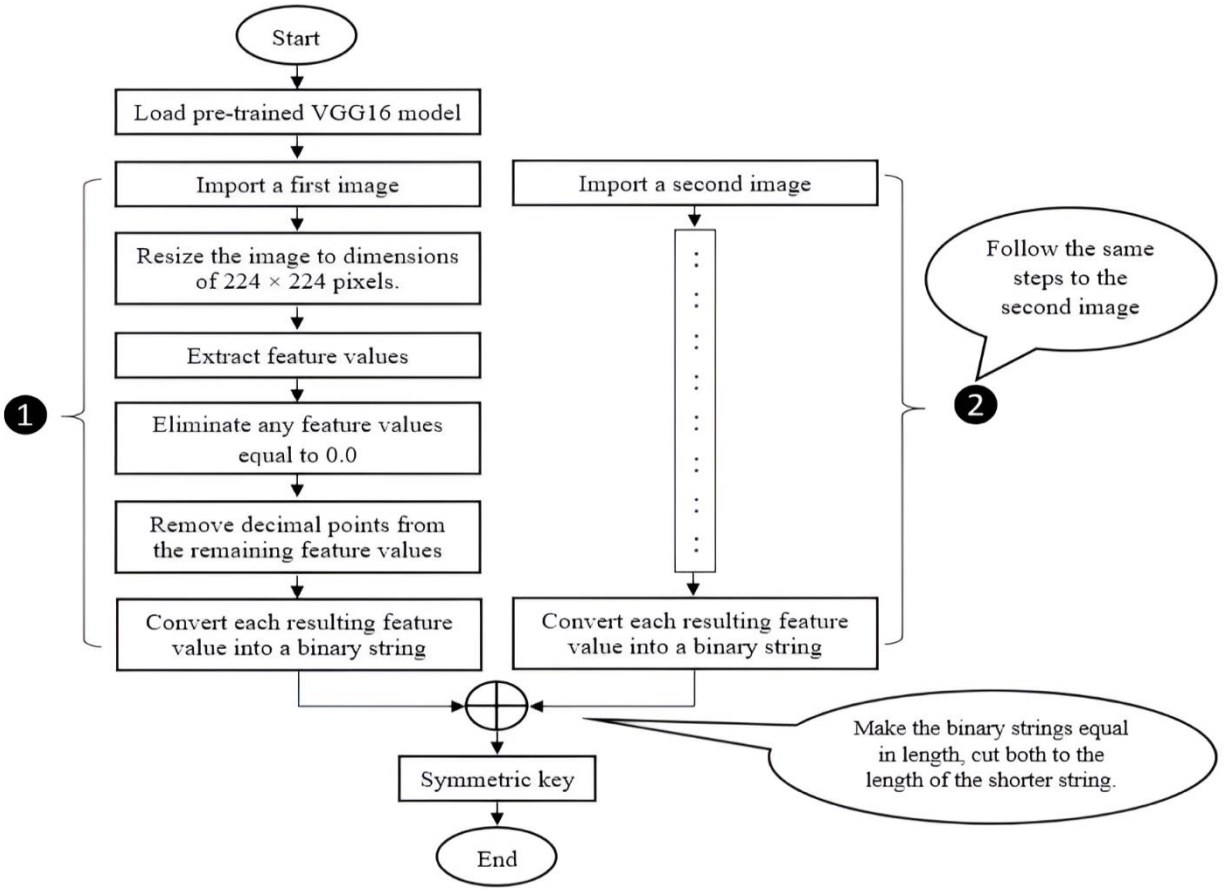


Fig. 4. Block diagram illustrating the proposed symmetric key generation method

Example

Here, a basic illustration of creating symmetric keys via the pretrained VGG16 model is presented.

- The selection of the first image is made randomly by the user

- The selection of the second image is made randomly by the user

- When features are extracted from the first image, since the image is in color, the output is represented as a three-dimensional array, as illustrated below.

[[[[0.0, 0.0, 0.0, 21.481338500976562, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 16.28042984008789, 0.0, 0.0, 0.0, 0.0, 27.702835083007812, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 8.304917335510254, 0.0, 0.0,

0.0, 0.0, 0.0, 0.0, 0.0, 2.6849257946014404, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 28.09803009033203, 0.0, 0.7807556390762329, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 11.043208122253418, 0.0, 0.0, 0.0, 0.0, 0.0, 0.029022634029388428, 10.636655807495117, …]]]]

- When features are extracted from the second image, since the image is in color, the output is represented as a three-dimensional array, as illustrated below.

[[[[0.0, 0.0, 7.432180881500244, 0.0, 0.0, 0.0, 13.72316837310791, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 8.511319160461426, 0.0, 0.0, 0.0, 0.0, 10.120380401611328, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 6.691157817840576, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.494135856628418, 0.0, 0.0, 0.0, 0.0, 1.4043443202972412, 0.0, 0.0, 0.0, 27.340776443481445, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 7.804123878479004, …]]]]

- Every occurrence of 0.0 is removed from the extracted features in the first image. The total number of features in this image is 3,192, as detailed below.

[21.481338500976562, 16.28042984008789, 27.702835083007812, 8.304917335510254, 2.6849257946014404, 28.09803009033203, 0.7807556390762329, 11.043208122253418, 0.029022634029388428, 10.636655807495117, 2.560093879699707, 8.163276672363281, 2.347464084625244, 21.656099319458008, 14.254408836364746, 17.918869018554688, …]

- Every occurrence of 0.0 is removed from the extracted features in the second image. The total number of features in this image is 4,840, as detailed below.

[7.432180881500244, 13.72316837310791, 8.511319160461426, 10.120380401611328, 6.691157817840576, 12.494135856628418, 1.4043443202972412, 27.340776443481445, 7.804123878479004, 13.260604858398438, 4.166805267333984, 7.468313694000244, 5.199467182159424, 7.691617965698242, 10.890722274780273, 6.505411624908447, …]

- All decimal points from the extracted features of the first image are eliminated, and then the modified values are transformed into a binary string. The final result consists of a binary string containing 169642 bits.

Feature	Remove the Dot (.)	Convert to Binary
21.481338500976562	21481338500976562	10011000101000100101001110111101010011010001111101100010
16.28042984008789	1628042984008789	10111001000101100100100101110111101101011000101010101
27.702835083007812	27702835083007812	11000100110101110010100111011000110010011010111010001000100
8.304917335510254	8304917335510254	11101100000010100011100111111101011000000110011101110
…	…	…

- All decimal points from the extracted features of the second image are eliminated, and then the modified values are transformed into a binary string. The final result consists of a binary string containing 257288 bits.

Feature	Remove the Dot (.)	Convert to Binary
7.432180881500244	7432180881500244	11010011001111000011101110101010110010010100001010100
13.72316837310791	1372316837310791	10011100000000111010110100110011011011100010100011111
8.511319160461426	8511319160461426	11110001111001111111111101011001000010001100001110010
10.120380401611328	10120380401611328	10001111110100011011101010111011010100011100100100000
…	…	…

- To make the two binary strings identical in length, they are shortened to match the length of the shorter string. Then, the XOR operation is applied between the binary representation of the first image and that of the second image. The resulting binary string consists of 169,642 bits, as demonstrated below.

[01001011100111001101000000101111000010001011101110000011001001000110001111110110111010000011010010100101000000011010010001101010010111100111010101010000011011000001000100010001100100100100100001111100111101110101101001111101000100010000100110011001000010011010110100101010111000010100110110110010000101010111001011110000101101010100011100111100011001010111001010101010010101111111100101000011001101101011011011010100000101110101101010110010111111100111011111000110110011010010111010010101001100010001110111, …]

5. IMAGE ALGORITHM SECURITY ANALYSIS

5.1. Encryption and Decryption Results

To validate the algorithm's effectiveness, experiments were performed using Lena, pepper, and flower images, each with a resolution of 512×512 pixels, in a Windows 11 environment with Python. The results, displayed in Figure 5, indicate that there is no distortion or data loss between the original and decrypted images. Conversely, the encrypted images completely obscure the features of the plaintext images. This demonstrates the algorithm's superior security performance.

Fig. 5. Encryption and decryption results.

5.2. Information Entropy

Entropy measures the level of randomness and unpredictability in an information source. It is calculated via the following formula:

$$H = -\sum_{i=0}^{L} p(i)\, log_2\, p(i) \tag{3}$$

where L represents the number of pixel values minus one, and p(i) is the probability of each pixel value occurring in the image. A higher entropy value signifies a more uniform distribution of pixel values. The optimal entropy value is 8.

TABLE V. INFORMATION ENTROPIES FOR ORIGINAL IMAGES OF VARYING SIZES AND THEIR CORRESPONDING ENCRYPTED IMAGES VIA THE PROPOSED LIGHTWEIGHT ENCRYPTION ALGORITHM

Image size	Image name	Original images			Encrypted images		
		Red	Green	Blue	Red	Green	Blue
256×256	Lena	6.9289	7.5891	7.2494	7.9971	7.9976	7.9974
	Flowers	7.6342	7.4183	7.5565	7.9976	7.9970	7.9976
	Pepper	7.1635	7.6458	7.3648	7.9973	7.9973	7.9970

512×512	Lena	6.9684	7.5940	7.2531	7.9992	7.9990	7.9992
	Flowers	7.6304	7.4252	7.5602	7.9991	7.9992	7.9992
	Pepper	7.1751	7.6341	7.3519	7.9986	7.9987	7.9987

In this experiment, various images of different sizes were used to assess entropy. Table V displays the information entropy values for the test images, both before and after encryption, via the proposed lightweight encryption algorithm. The original images exhibit relatively low entropy. However, the entropy values for the encrypted images are nearly 8, indicating a highly uniform pixel distribution and rendering any information from pixel patterns unattainable.

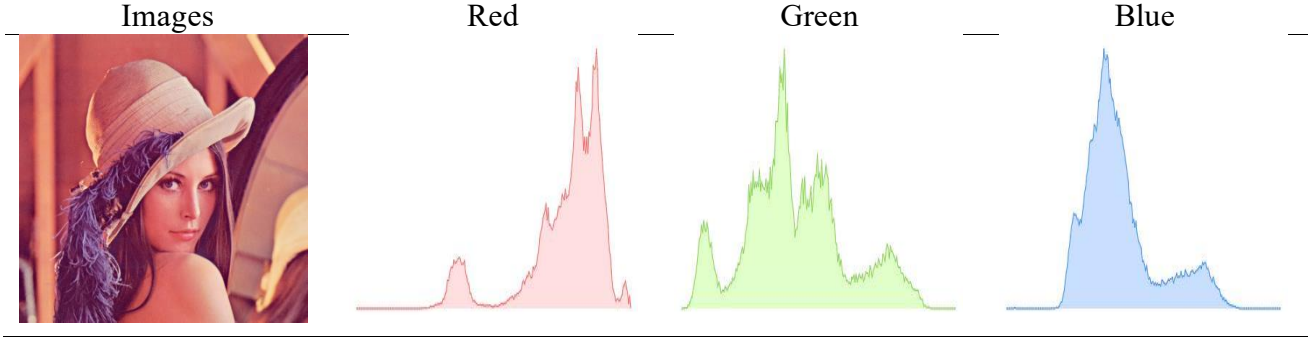
TABLE VI. THE INFORMATION ENTROPIES OF CIPHERED IMAGES GENERATED BY DIFFERENT IMAGE ENCRYPTION ALGORITHMS WERE EVALUATED VIA THE LENA TEST IMAGE

Encryption algorithm	**Image size**	**Encrypted images**		
		Red	**Green**	**Blue**
Proposed	256× 256	7.9971	7.9976	7.9974
Ref.[7], hybrid 2D hyperchaos and genetic recombination		7.9971	7.9970	7.9970
Ref. [8], four-wing chaotic and compressive sensing		7.9972	7.9967	7.9967
Ref.[9], hybrid 2D chaotic map		7.9872 (average)		
Ref. [32], 2DNA encoding and chaotic 2D logistic map		7.9898 (average)		
Ref. [16], NCA map-based CML and one-time keys		7.9892	7.9896	7.9896
Ref. [17], lossless DNA encryption scheme		7.9895	7.9894	7.9894
Ref. [18], integrated bit-level permutation		7.9943	7.9943	7.9942
Ref. [19], 4-pixel Feistel structure and multiple chaotic		7.9913	7.9914	7.9916
Ref. [20], hybrid genetic algorithm and a DNA sequence		7.9896	7.9893	7.9907
Ref. [21], deduced gyrator transform		7.9899	7.9873	7.987
Proposed	512× 512	7.9992	7.9990	7.9992
Ref. [8], four-wing chaotic and compressive sensing		7.9991	7.9991	7.9992
Ref. [22], generalized heat equation associated with generalized Vigene`re type table over symmetric group		7.9912	7.9914	7.9915
Ref. [23], skew tent map and hyper chaotic system of 6th-order CNN		7.9278	7.9744	7.9705
Ref. [24], deep learning and block embedding		7.9916	7.9913	7.9919
Ref. [25], block scrambling and chaos		7.9974	7.9976	7.9974
Ref. [26], 2DNLCML system and genetic operations		7.9917	7.9912	7.9918

We also compared the information entropy of images encrypted via various algorithms, as shown in Table VI. The images encrypted with the newly proposed lightweight encryption algorithm have higher average entropy values than those encrypted by other methods, indicating the improved effectiveness of our approach.

5.3 Histogram Analysis

The pixel distribution of an image is represented by its histogram. An eavesdropper might attempt to compromise the encrypted image via histogram analysis. To counteract such statistical attacks, the histogram of the cipher image must be as uniform as possible. Figure 6 shows different original images of size 512×512 alongside the histograms of their R, G, and B channels both before and after encryption. The histograms after encryption appear more uniform than the original histograms. Consequently, the system demonstrates robustness against histogram attacks.



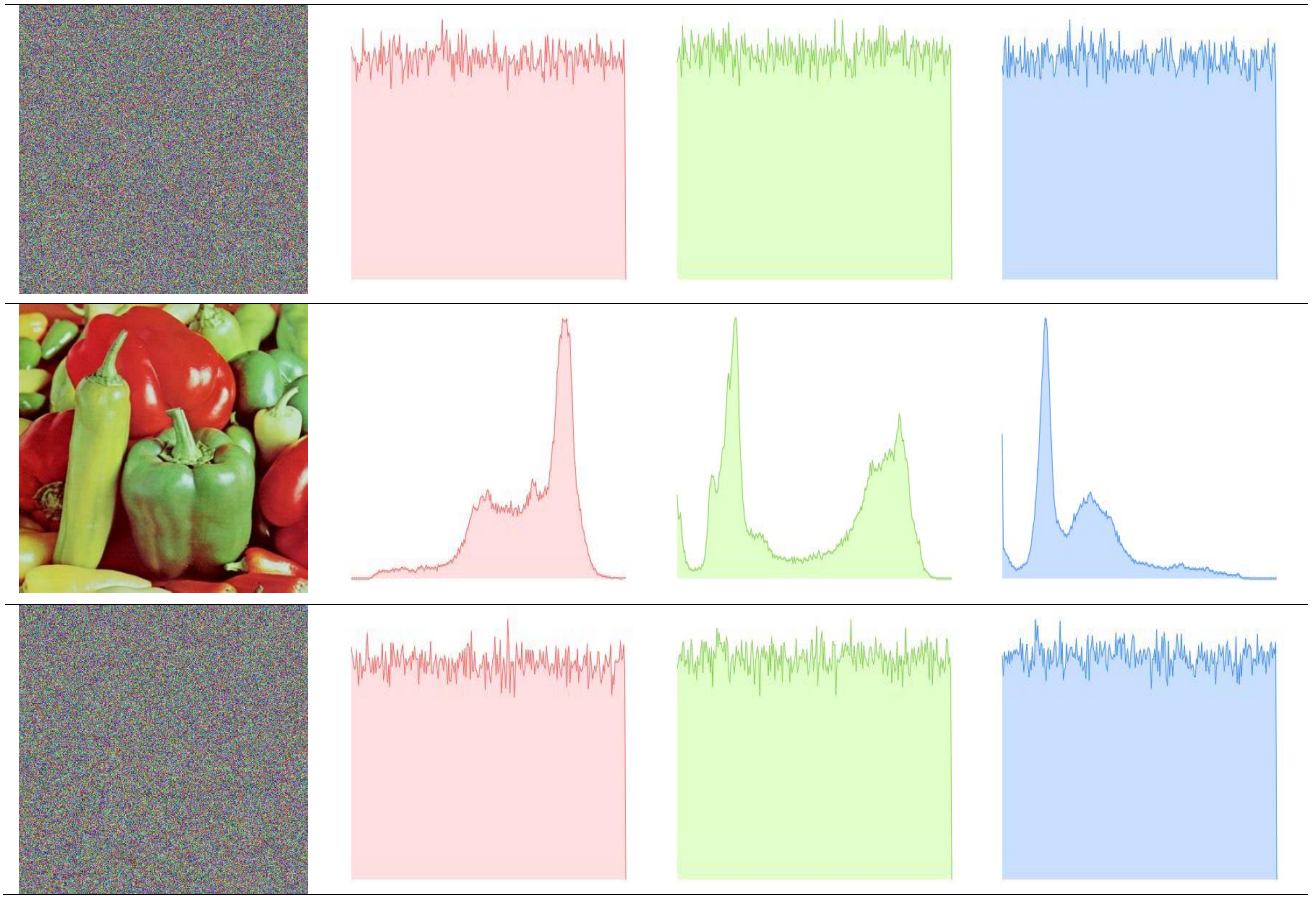


Fig. 6. Histogram of images

In the histogram of a color image of size 512×512 (both original and encrypted), the X-axis represents the pixel intensity values ranging from 0 to 255 for each color channel (Red, Green, and Blue), whereas the Y-axis indicates the frequency of pixels for each intensity level, varying from 0 to a maximum of 262,144 pixels (since the total number of pixels in the image is 512×512). The histogram of the original image typically shows peaks and variations corresponding to the image content, whereas the histogram of the encrypted image tends to be more uniform, indicating a better distribution of pixel intensities due to encryption.

5.4 Correlation Analysis

Correlation represents the relationship between neighboring pixels. In typical images, the information is highly redundant, leading to a strong correlation between adjacent pixels. An effective encryption algorithm aims to decrease this correlation, ideally bringing it to 0. The correlation calculation is demonstrated in Eq. (4).

$$\begin{cases} C_{xy} = \dfrac{\frac{1}{N}\sum_{i=1}^{n}\big(x_i - E(x)\big)\big(y_i - E(y)\big)}{\sqrt{D(x)}\sqrt{D(y)}} \\ E(X) = \dfrac{1}{N}\sum_{i=1}^{n} x_i \quad , \quad D(x) = \dfrac{1}{N}\sum_{i=1}^{n}(x_i - E(x))^2 \end{cases} \tag{4}$$

Here, xi and yi represent the pixel values of adjacent pixels, N is the number of pixels, and (x) and (y) denote the mean values of xi and yi, respectively. Eq. (4) was used to determine the image correlation, as presented in Table VII. Figures 7–9 display the correlations for the red, green, and blue channels for Lena with a size of 512×512.

TABLE VII. THE RESULTS OF THE CORRELATION ANALYSIS

Image	channel	H	V	D
Plain image of lena	Red	0.9694	0.9726	0.9434
	Green	0.9672	0.9782	0.9496
	Blue	0.9276	0.9492	0.9055
Encrypted image of lena	Red	-0.0027	-0.0011	0.0019
	Green	-0.0027	-0.0015	0.0020
	Blue	0.0023	0.0031	0.0025
Ref. [6]	Red	0.0040	–0.0012	0.0113
	Green	–0.0013	0.0079	0.0037
	Blue	0.0025	–0.0007	0.0021
Ref. [7]	Red	−0.0035	0.0034	−0.0003
	Green	−0.0095	0.0051	0.0026
	Blue	−0.0042	−0.0016	0.0006
Ref. [27]	Red	– 0.0064	0.0053	0.0061
	Green	0.0018	–0.0047	0.0027
	Blue	0.0099	0.0043	0.0035
Ref. [24]	Red	−0.0046	0.0072	0.0009
	Green	−0.0015	0.0056	−0.0125
	Blue	−0.0091	−0.0076	−0.0145

Table VII shows that the original image has a correlation coefficient close to 1, reflecting strong correlation and redundancy. On the other hand, the encrypted image has a correlation coefficient near 0, indicating that the proposed algorithm successfully disrupts the correlation between neighboring pixels.

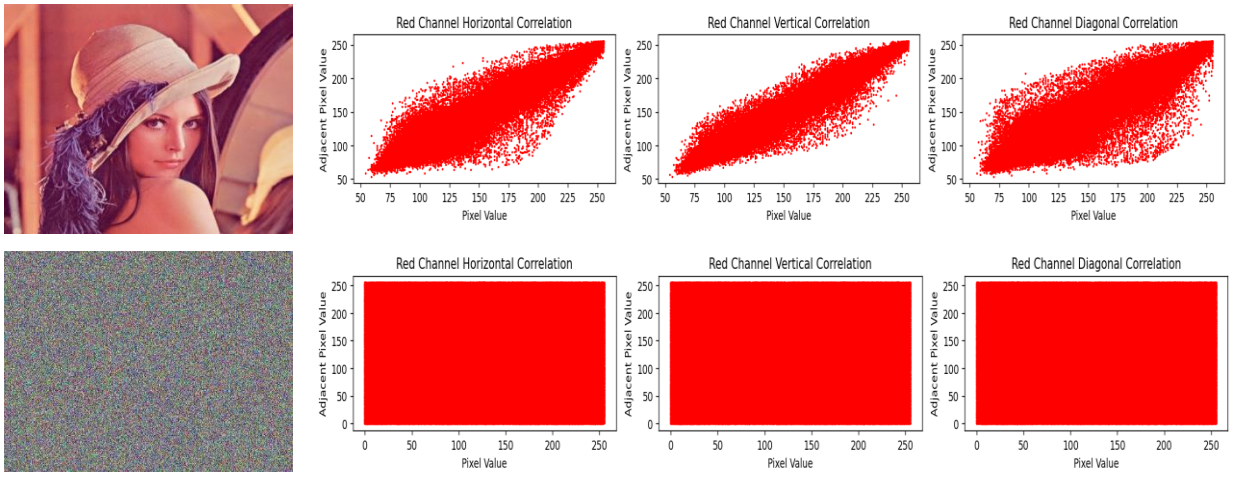


Fig. 7. R-channel correlation analysis for the original image and encrypted image (horizontal, vertical, and diagonal)

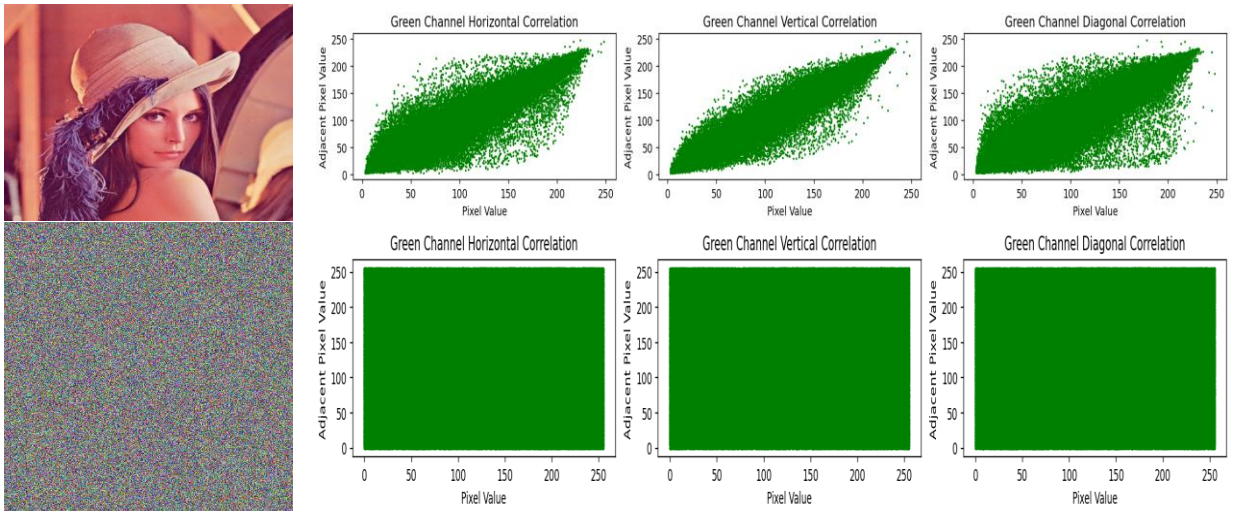


Fig. 8. G-channel correlation analysis for the original image and encrypted image (horizontal, vertical, and diagonal)

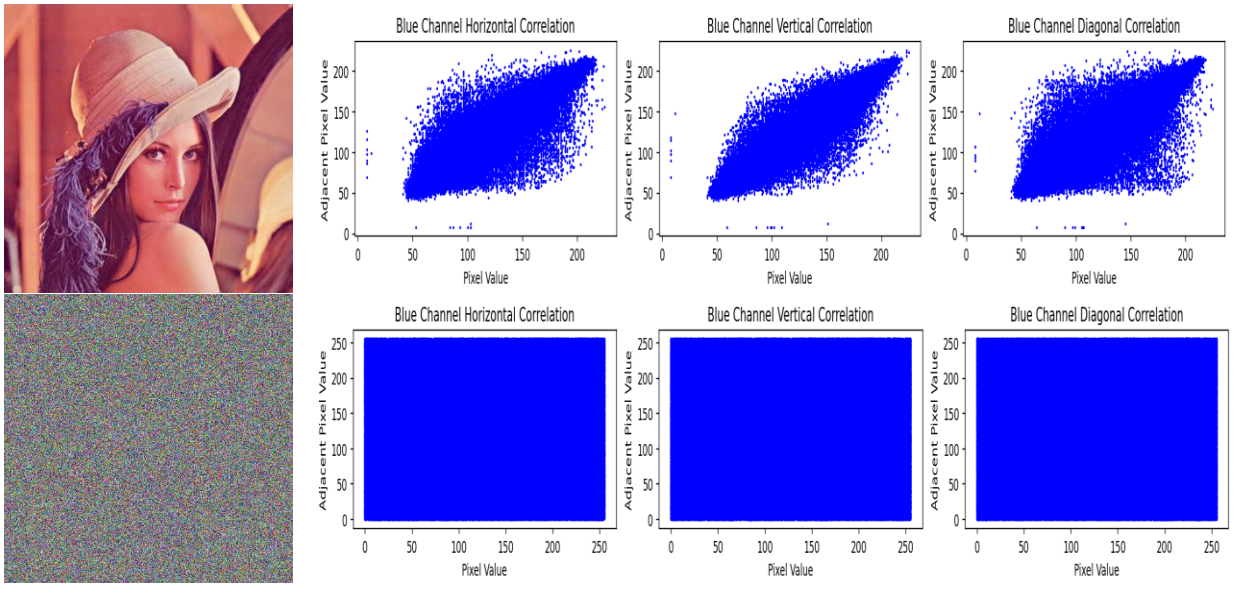


Fig. 9. B-channel correlation analysis for the original image and encrypted image (horizontal, vertical, and diagonal)

Figures 7–9 illustrate that the pixels in the original image are primarily concentrated along the diagonal, exhibiting high density and strong correlation. In the encrypted image, the pixels are evenly dispersed, indicating a weak correlation. This confirms that the encrypted image has a low correlation.

5.5 Differential Attack Analysis

A differential attack is another common form of attack in which attackers start with a base image and make minor modifications, such as changing a single bit [31]. They then compared the ciphertexts of the original and modified images. Effective encryption algorithms should exhibit high sensitivity to changes in the plaintext, resulting in significant alterations in the encrypted image even with slight modifications to the plaintext. This sensitivity is typically evaluated via metrics such as the number of pixel change rates (NPCRs) and the unified average change intensity (UACI), which can be calculated via the formulas presented in Eq. (5).

$$\begin{cases} NPCR = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\% \\ UACI = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N} |c_1(i,j) - c_2(i,j)|}{M \times N \times 255} \times 100\% \\ D(i,j) = \begin{cases} 1, & if\ c_1(i,j) \neq c_2(i,j) \\ 0, & if\ c_1(i,j) = c_2(i,j) \end{cases} \end{cases} \quad (5)$$

In this context, c_1 and c_2 represent the cipher images generated by the proposed method, with M and N denoting the number of pixels in the image rows and columns.

When the NPCR value exceeds 0.996, the UACI value falls between 0.33329 and 0.335541. The optimal values for the NPCR and UACI are 99.6094% and 33.4635%, respectively [6].

TABLE VIII. NPCR AND UACI OF THE ENCRYPTED LENA IMAGE VIA THE PROPOSED LIGHTWEIGHT ALGORITHM

Image	Image size	NPCR (%)			UACI (%)		
		Red	Green	Blue	Red	Green	Blue
proposed method	512×512	99.6048	99.6090	99.6014	33.3680	33.4909	33.4099
Ref. [6]		99.6136	99.5922	99.6109	33.4783	33.4769	33.4916
Ref. [7]		99.6101	99.6063	99.6014	33.4234	33.5112	33.5513
Ref. [28]		99.6016	99.6205	99.6095	33.2483	33.4977	33.3877
Ref. [29]		99.6056	99.6147	99.6235	33.4108	33.4653	33.4901
Ref. [30]		99.6069	99.6102	99.5921	33.4926	33.4620	33.4961
Proposed method	256×256	99.6185	99.6185	99.6033	33.3832	33.4089	33.4008
Ref. [9]		99.502 (average)			33.6542 (average)		
Ref. [10]		99.61 (average)			33.69 (average)		
Ref. [32]		99.5519 (average)			32.8111 (average)		

As shown in Table VIII, the NPCR and UACI values achieved by this algorithm meet the standards of [6, 35]. The NPCR and UACI test results from other studies also fall within the acceptable range. Compared with the other algorithms, this algorithm's NPCR and UACI values are closer to the ideal values, indicating that the encrypted image is highly sensitive to plaintext information. This heightened sensitivity is advantageous for defending against differential attacks.

5.6 Efficiency of the F-Function

The F-Function plays a crucial role in the proposed lightweight encryption algorithm, enhancing its security by improving diffusion, confusion and randomness in the encrypted image. It introduces complex transformations to pixel values, ensuring that the encryption process effectively disguises the statistical patterns present in the original image.

TABLE IX. INFORMATION ENTROPY COMPARISON FOR THE ORIGINAL LENA IMAGES, THEIR CORRESPONDING ENCRYPTED IMAGES VIA THE PROPOSED LIGHTWEIGHT ENCRYPTION ALGORITHM, AND THE ENCRYPTED IMAGES AFTER REMOVING THE F-FUNCTION

Image size	Entropy with F-Function			Entropy without F-Function		
	Red	Green	Blue	Red	Green	Blue
256×256	7.9971	7.9976	7.9974	7.9967	7.9970	7.9968
512×512	7.9992	7.9990	7.9992	7.9822	7.9496	7.9323

TABLE X. NPCR COMPARISON OF THE ENCRYPTED LENA IMAGE USING THE PROPOSED LIGHTWEIGHT ENCRYPTION ALGORITHM AND THE ENCRYPTED IMAGE AFTER REMOVING THE F-FUNCTION

Image size	NPCR(%) with F-Function			NPCR(%) without F-Function		
	Red	Green	Blue	Red	Green	Blue
256×256	99.6185	99.6185	99.6033	98.3041	98.3325	98.3127
512×512	99.6048	99.6090	99.6014	98.6912	98.6755	98.6887

TABLE XI. UACI COMPARISON OF THE ENCRYPTED LENA IMAGE USING THE PROPOSED LIGHTWEIGHT ENCRYPTION ALGORITHM AND THE ENCRYPTED IMAGE AFTER REMOVING THE F-FUNCTION

Image size	UACI (%) with F-Function			UACI (%) without F-Function		
	Red	Green	Blue	Red	Green	Blue
256×256	33.3832	33.4089	33.4008	12.6390	12.4113	12.4782
512×512	33.3680	33.4909	33.4099	12.5034	12.4911	12.5379

TABLE XII. CORRELATION ANALYSIS COMPARISON FOR THE ORIGINAL IMAGES, THEIR CORRESPONDING ENCRYPTED IMAGES VIA THE PROPOSED LIGHTWEIGHT ENCRYPTION ALGORITHM, AND THE ENCRYPTED IMAGES AFTER REMOVING THE F-FUNCTION

Image	channel	H	V	D
Encrypted image of Lena with F-Function	Red	-0.0027	-0.0011	0.0019
	Green	-0.0027	-0.0015	0.0020
	Blue	0.0023	0.0031	0.0025
Encrypted image of Lena without the F-Function	Red	-0.0021	**-0.0040**	**0.1568**
	Green	**0.0030**	-0.0022	**0.0600**
	Blue	0.0013	-0.0001	**0.0286**

Table IX presents the entropy values for encrypted images of different sizes, both with and without the F-Function. The results show that the entropy values decrease slightly for the 256×256 image when the F-Function is removed. However, for 512×512 images, the entropy decreases significantly when the F-Function is removed, especially in the green and blue channels. This finding indicates that the F-Function plays a crucial role in maintaining high randomness, ensuring stronger encryption security for larger images.

In Table X, the results show that with the F-Function, the NPCRs are consistently high (above 99.60%). However, when the F-Function is removed, the NPCR decreases noticeably, especially for 512×512 images. This reduction suggests that the F-Function significantly enhances sensitivity to small changes, making it harder for attackers to predict the encrypted output.

In Table XI, the results show that with the F-Function, the UACI values are consistently approximately 33%, ensuring that a high level of pixel intensity changes. However, when the F-Function is removed, the UACI values drop drastically to around approximately 12.5%, representing an ~62% decrease in the encryption strength. This significant decline confirms that the F-Function plays a vital role in effectively spreading pixel modifications, increasing resistance to differential attacks.

In Table XII, the results show that with the F-Function, the horizontal (H), vertical (V), and diagonal (D) correlation values are near zero, reflecting a well-scrambled image. However, when the F-Function is removed, the diagonal correlation (D) increases dramatically, particularly in the Red channel (from 0.0019 to 0.1568). This increase suggests that adjacent pixels in the encrypted image become more predictable without the F-Function, reducing encryption security.

6. CONCLUSIONS

This paper presents a newly developed lightweight symmetric encryption algorithm that features an entirely new F-function specifically designed to enhance diffusion and confusion. In this proposed structure of new lightweight algorithms for color image encryption, my initial attempts aimed to successfully meet all evaluation criteria (entropy, correlation coefficient, NPCR, UACI) for cipher images generated from encrypted images. Removing the F-Function from the proposed structure reduced the randomness of the encryption process by approximately 30–40%, underscoring its essential role in achieving high randomness in the encrypted image. Additionally, a unique method using the Gauss map to generate shift values further strengthens confusion within the block cipher. These findings confirm that the proposed algorithm achieves robust encryption and high resistance to differential attacks.

Conflicts of interest
The authors declare that they have no conflicts of interest.

Funding
The authors received no specific funding for this study.

Data availability
Data are contained within the article.

References
[1] M. M. Abd Zaid, and S. Hassan, "Proposal Framework to Lightweight Cryptography Primitives", *Engineering and Technology Journal*, Vol. 40, pp. 516-526, 2022.

[2] A. R. Alawi, and N. F. Hassan, "A Proposal Video Encryption Using Light Stream Algorithm", *Engineering and Technology Journal*, Vol. 39, pp. 184-196, 2021.

[3] R. W. Abd Aljabar, and N. F. Hassan, "Encryption VoIP based on Generated Biometric Key for RC4 Algorithm", *Engineering and Technology Journal*, Vol. 39, pp. 209-221, 2021.

[4] M. A. Jumaah, Y. H. Ali, T. A. Rashid, and S. Vimal, "FOXANN: A Method for Boosting Neural Network Performance", *Journal of Soft Computing and Computer Applications*, Vol. 1, 1001, 2024.

[5] Y. Cao, and Y. Song, "Color Image Encryption Based on an Evolutionary Codebook and Chaotic Systems", *Entropy*, Vol. 26, No. 7, 597, 2024.

[6] J. Xu, B. Zhao, and Z. Wu, "Research on Color Image Encryption Algorithm Based on Bit-Plane and Chen Chaotic System", *Entropy*, Vol. 24, No. 2, 186, 2022.

[7] Y. Xu, J. Liu, Z. You, and T. Zhang, "A Novel Color Image Encryption Algorithm Based on Hybrid Two-Dimensional Hyperchaos and Genetic Recombination", *Mathematics*, Vol. 12, No.22, 3457, 2024.

[8] L. Zhang, and X. L. An, "Dynamic Analysis of a Four-Wing Chaotic System and Application in Image Encryption Based on Compressive Sensing", *IEEE Access*, Vol. 12, pp. 2573-2588, 2024.

[9] B. A. Hameed, and E. K. Gbashi, "New S-Box Generation Based on Hybrid Two-Dimensional Chaotic Map for Color Image Encryption", *International Journal of Intelligent Engineering and Systems*, Vol.17, No.6, pp. 702-716, 2024.

[10] S. M. Ali, A. H. Zwiad, R. M. Al-Amri, and A. K. Farhan, "Generation of Dynamic Substitution Boxes Using HSM Chaos System for Application in Color Images Encrypting", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 6, pp. 519-530, 2023.

[11] X. Li, J. Zeng, Q. Ding, and C. Fan, "A Novel Color Image Encryption Algorithm Based on 5-D Hyperchaotic System and DNA Sequence", *Entropy*, Vol. 24, No. 9, 1270, 2022.

[12] H. K. Sarmah, M. C. Das, T. K. Baishya, and R. Paul, "Characterising Chaos In Gaussian Map", *International Journal of Advanced Scientific and Technical Research*, Vol. 1, pp. 160-172, 2016.

[13] O. Camps, S. G. Stavrinides, C. D. Benito, and R. Picos, "Implementation of the Hindmarsh–Rose Model Using Stochastic Computing", *Mathematics*, Vol. 10, No. 23, 4628, 2022.

[14] A. T. Khudhair, A. T. Maolood, and E. K. Gbashi, "A Novel Approach to Generate Dynamic S-Box for Lightweight Cryptography Based on the 3D Hindmarsh Rose Model", *Journal of Soft Computing and Computer Applications*, Vol. 1, 1003, 2024.

[15] A. T. Khudhair, A. T. Maolood, and E. K. Gbashi, "Symmetric Keys for Lightweight Encryption Algorithms Using a Pre–Trained VGG16 Model", *Telecom*, Vol. 5, No. 3, pp. 892-906, 2024.

[16] W. Xiangjun, W. Kunshu, W. Xingyuan, K. Haibin, and J. Kurths, "Color image DNA encryp tion using NCA map-based CML and one-time keys", *Signal Processing*, Vol. 148, pp. 272-287, 2018.

[17] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images", *Multimedia Tools and Applications*, Vol. 77, pp. 12349-12376, 2018.

[18] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation", *Multimedia Tools and Applications*, Vol. 77, pp. 6883-6896, 2018.

[19] W. Yao, X. Zhang, Z. Zheng, and W. Qiu, "A colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems", *Nonlinear Dynamics*, Vol. 81, pp. 151-168, 2015.

[20] R. Enayatifar, A. H. Abdullah, and I. F. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence", *Optics and Lasers in Engineering*, Vol. 56, pp. 83-93, 2014.

[21] L. Yao, C. Yuan, J. Qiang, S. Feng, and S. Nie, "An asymmetric color image encryption method by using deduced gyrator transform", *Optics and Lasers in Engineering*, Vol. 89, pp. 72-79, 2017.

[22] M. Kumar, G. Sathish, M. Alphonse, and R. A. M. Lahcen, "A new RGB image encryption using generalized heat equation associated with generalized Vigene`re type table over symmetric group", *Multimedia Tools and Applications*, Vol. 78, pp. 28025-28061, 2019.

[23] A. Kadir, A. Hamdulla, and W. Q. Guo, "Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN", *Optik*, Vol. 125, pp. 1671-1675, 2014.

[24] Y. Liu, G. Cen, B. Xu, and X. Wang, "Color Image Encryption Based on Deep Learning and Block Embedding", *Security and Communication Networks*, 6047349, 2022.

[25] K. M. Hosny, S. T. Kamal, and M. M. Darwish, "A color image encryption technique using block scrambling and chaos", *Multimedia Tools and Applications*, Vol. 81, pp. 505–525, 2022.

[26] Y. Q. Zhang, Y. He, P. Li, and X. Wang, "A new color image encryption scheme based on 2DNLCML system and genetic operations", *Optics and Lasers in Engineering*, Vol. 128, 106040, 2020.

[27] H. R. Amani, and M. Yaghoobi, "A New Approach in Adaptive Encryption Algorithm for Color Images Based on DNA Sequence Operation and Hyper-Chaotic System", *Multimedia Tools and Applications*, Vol. 78, pp. 21537–21556, 2019.

[28] P. Li, J. Xu, J. Mou, and F. Yang, "Fractional-order 4D hyperchaotic memristive system and application in color image encryption", *EURASIP journal on image and video processing*, 2019.

[29] E. Hasanzadeh, and M. Yaghoobi, "A Novel color image encryption algorithm based on substitution box and hyper chaotic system with fractal keys", *Multimedia Tools and Applications*, Vol. 79, pp. 7279–7297, 2020.

[30] B. Arpacı, E. Kurt, K. Çelik, and B. Ciylan, "Colored Image Encryption and Decryption with a New Algorithm and a Hyperchaotic Electrical Circuit", *Journal of Electrical Engineering & Technology*, Vol. 15, pp. 1413-1429, 2020.

[31] A. T. Khudhair, A. T. Maolood, and E. K. Gbashi, "Symmetry Analysis in Construction Two Dynamic Lightweight S-Boxes Based on the 2D Tinkerbell Map and the 2D Duffing Map", *Symmetry*, Vol. 16, No. 7, 872, 2024.

[32] A. H. Alrubaie, M. A. A. Khodher, and A. T. Abdulameer, "Image encryption based on 2DNA encoding and chaotic 2D logistic map", *Journal of Engineering and Applied Sciece*, Vol. 70, pp. 1-21, 2023.

[33] A. A. Abbod, A. K. A. Hassan, and M. Alqaseer, "Analyzing user behavior for targeted commercial advertisements using Apriori and K-means algorithms", AIP *Conf. Proc,* Vol. 3169, No. 1, 2025.

[34] S. M. Shareef and R. F. Hassan, "Improved Blockchain Technique based on Modified SLIM Algorithm for Cyber Security", *Mesopotamian Journal of CyberSecurity*, Vol.5, No.1, pp.147-164, 2025.

[35] J. Ayad, N. Qaddoori, and H. Maytham, "Enhanced Audio Encryption Scheme: Integrating Blowfish, HMAC-SHA256, and MD5 for Secure Communication", *Mesopotamian Journal of CyberSecurity*, Vol. 5, No. 1, pp. 178-186, 2025.