Iraqi Journal of Science, 2023, Vol. 64, No. 12, pp: 6534- 6547
DOI: 10.24996/ijs.2023.64.12.35

ISSN: 0067-2904

A Hybrid Algorithms Based on the Aizawa Attractor and Rabbit-Lightweight Cipher for Image Encryption

Mohammed Gheni Alwan*, Enas Tariq Khudair, Ekhlas Falih Naser
Computer Sciences Dept., Univ. of Technology, Baghdad, Iraq

Received: 2/5/2022 Accepted: 28/10/2022 Published: 30/12/2023

Abstract

Social media and networks rely heavily on images. Those images should be distributed in a private manner. Image encryption is therefore one of the most crucial components of cyber security. In the present study, an effective image encryption technique is developed that combines the Rabbit Algorithm, a simple algorithm, with the Attractor of Aizawa, a chaotic map. The lightweight encryption algorithm (Rabbit Algorithm), which is a 3D dynamic system, is made more secure by the Attractor of Aizawa. The process separates color images into blocks by first dividing them into bands of red, green, and blue (RGB). The presented approach generates multiple keys, or sequences, based on the initial parameters and conditions, which are then utilized for encrypting each one of the blocks. The peak signal-to-noise ratio (PSNR), Mean Squared Error (MSE), and structural similarity index measure (SSIM) tests are among performance tests that are used to assess the effectiveness of the approach. With the help of National Institute of Standards and Technology (NIST), the produced keys have been checked for randomness. Additionally, research is being done into how long decryption and encryption take. The encryption approach is particularly secure because of the chaotic attractor's unpredictability. The correlation coefficient, Shannon entropy, and a histogram are used to analyse the encryption technique's performance. The findings demonstrate that the suggested encryption approach dependably operates since it makes use of a mechanism to strengthen security against the attack of known or selected plain-text. The outcomes also demonstrated that both decoding and encoding using the suggested encoding approach require less time. For instance, images 3's decoding time is 0.002000, whereas images 3's encoding time is 0.001200.

Keywords: Security, Key Stream, Entropy, 3D Chaotic System, Images encryption

الخوارزميات الهجينة المعتمدة على Aizawa Attractor وتشفير الأرنب الخفيف لتشفير الصور

محمد غني علوان* , أيناس طارق خضير , أخلاص فالح ناصر
قسم علوم الحاسوب, الجامعة التكنولوجية, بغداد, العراق

الملخص

تعتبر الصور من المكونات المهمة لوسائل التواصل الاجتماعي والشبكات. لذلك فان التوزيع السري لهذه الصور أمر لا بد منه. وبالتالي، فإن أحد أهم جوانب الأمن السيبراني هو تشفير الصور. في هذا البحث تم اقتراح طريقة فعالة لتشفير الصور تجمع بين خوارزمية خفيفة الوزن (خوارزمية الأرنب) مع خريطة فوضوية (**Aizawa Attractor**). يضيف ***Aizawa Attractor***، وهو نظام ديناميكي ثلاثي الأبعاد، السرية إلى

*Email: Mohammed.G.Alwan@uotechnology.edu.iq

خوارزمية التشفير خفيفة الوزن (خوارزمية الأرنب). تقسم الطريقة الصور الملونة إلى نطاقات أحمر وأخضر وأزرق (***RGB***)، والتي يتم تقسيمها بعد ذلك إلى كتل. يتم توليد المفاتيح المتعددة، أي التسلسلات، بالطريقة المقترحة وفقًا للشروط والمعلمات الأولية وتستخدم لتشفير كل كتلة. يتم تقييم الطريقة من خلال العديد من اختبارات الأداء، بما في ذلك متوسط الخطأ التربيعي (***MSE***)، ونسبة ذروة الإشارة إلى الضوضاء (***PSNR***) وقياس مؤشر التشابه الهيكلي (***SSIM***). تم أيضًا اختبار المفاتيح التي تم إنشاؤها من أجل العشوائية باستخدام المعهد الوطني للمعايير والتكنولوجيا (***NIST***). علاوة على ذلك، يتم أيضًا التحقق من الوقت اللازم للتشفير وفك التشفير. عدم القدرة على التنبؤ بالجاذب الفوضوي يجعل طريقة التشفير آمنة للغاية. يتم تحليل أداء طريقة التشفير عن طريق الرسم البياني، ومعامل الارتباط، وانتروبيا شانون. أظهرت النتائج أن طريقة التشفير المقترحة تعمل بشكل موثوق لأنها تستعمل لتعزيز الأمان مقابل هجوم النص العادي المختار أو المعروف. أظهرت النتائج أيضًا أن طريقة التشفير المقترحة تستغرق وقتًا أقل للتشفير وفك التشفير، على سبيل المثال في الصورة 3، فأن الوقت المستغرق للتشفير هو 0.001200 بينما وقت فك التشفير هو 0.002000.

1. Introduction

As an outcome of a fast growth in a communication and digital information, security has been became an integral component of digital media. Images, videos, and audio are exchanged and disseminated in a variety of areas. For example, the public uses digital communication for financial transactions and commercial interactions; the government uses it to communicate private material; and the medical profession uses it for patient records. All of these forms of communication require user authentication as well as dependable data transmission, and encryption methods are valuable tools for achieving these goals [1]. The two primary groups of cryptosystems are stream ciphers and block ciphers. The methods used to transform plain communications into encrypted messages vary across the two groups.

A stream cipher algorithm encrypts the message bit by bit using a secret key generating mechanism and then decrypts it using the same encryption approach and secret key generator [2]. The encryption key can be generated using a variety of approaches, such as clock-controlled generators, non-linear combination generators, shift-registers, and so on. In contrast to stream encryption, a block cipher transforms an entire message block at the same time [3].

Rivest Cipher 5 (RC5) and Data Encryption Standard (DES) [6] are two of the most widely utilized block cipher algorithms, whilst Chacha and Salsa [4] and Rabbit [5] are two regularly utilized stream algorithms. Another way of classification separates cryptosystems into those with private-type (symmetric) keys and those with public-type (asymmetric) keys. The receiver and transmitter in private-key cryptography use one key for decryption and encryption, respectively. A public key is used for the plain-text encryption, whereas a private key is deployed for decrypting it in the case when the public-key cryptography is employed [7].

Since the 1990s, chaos theory has been used in a variety of fields, including engineering, physics, economics, and biology, with various academics and scientists conducting extensive research on the chaotic systems based on this theory. A strong link has been discovered between cryptography and chaos theory [8]. Lightweight Cryptography (LWC) [9] focuses on cryptographic techniques appropriate for constrained environments, including sensors, RFID tags, and contactless smart cards. Lightweight block and stream ciphers, which are mainly used for confidentiality and data integrity, should work well in embedded systems [10].

Block chippers are the most commonly used option in LWC because their designs are typically simpler and their safety characteristics have been extensively researched. One of these block chippers is used for Rabbit stream encryption and features a secret key for an endless chip stream. The encryption keystream is then Exclusive-OR (Ex-OR) Gated with the data stream.

Thus, we created an identical chip keystream and EX-OR'd it with the input chip stream to retrieve the data stream. Nice, simple carriers enhance the general algorithm's power to spin and partial blockage variations [11].

Chaotic maps are ideal for the creation of cipher algorithms. The Aizawa Chaotic technique is one of these approaches. Random ciphers are typically simple to build, fast, and resource-efficient, making multimedia data encryption techniques extremely efficient and secure [12].

A hybrid encryption technique is suggested in the presented work. Sections 2, 3, and 4 contain background information on the system. Section 5 provides the suggested system's steps and structure. Also, section 6 is used for testing. In section 7, conclusions are offered.

2. Literature review

The study that has been done into the Aizawa Attractor and Rabbit-Lightweight algorithms for image encryption is reviewed below. A novel hybrid chaotic map was built to encrypt and hide images. Message Digest method 5 (MD5) was used as a starting condition and control parameter to throw off the trajectory and improve security against plaintext-chosen and differential attacks. A ciphered image was put into numerous carrier images to reduce suspicion and increase resilience. The data revealed that the proposed procedure is exceedingly safe, dependable, and effective [3].

By combining the idea of steganography with the concepts of image compression and cryptography, a high degree of security is achieved. The Aizawa attractor map was used to create chaos-based image encryption. The protected stego-image creator and image were developed for giving a high security level while using minimal memory space for storage. The security of the encrypted image was determined using a statistical method that uses the correlation functions between two neighboring pixels, followed by a differential attack and key space analysis. This method attempts to address issues with encryption, such as a lack of security and limited key space [14].

[15] Used a new color scheme for image encryption depending on order of fractional hyper chaotic framework. Also, a framework that treats the system's parameters and a starting value as secret keys generates four chaotic sequences. A plain-image is simultaneously encrypted with the use of the (XOR) and shuffling processes. Security investigations, including study of correlation, histogram analysis, differential attacks, and analysis of key sensitivity, were used to thoroughly examine a proposed encryption approach. This encryption technique is safe and suitable for color image encryption because of its wide key space, great sensitivity to key features, and resistance to differential attacks. In [16], a hyper chaotic map is used for handling the sensitivity of the starting conditions, the pseudo-randomness regarding the chaotic maps, and the chaotic control parameters, and the image is encrypted with the use of a symmetric ChaCha stream-cipher. Security is improved by leveraging the initial seed number, parameter variability, and introducing unexpected directions. The broad key space provides a proposed lightweight for encryption of an image with resistance to brute-force attacks. Furthermore, based on histogram, correlation, and entropy constraints, the used light-weight encryption for an image is resistant to statistical cracking and image insecurity.

The novel use of an Aizawa chaotic map to solve the security and performance issues that many current permutation-diffusion image ciphers suffer from improves security against known/chosen plaintext attacks. The suggested system has been put to the test, and the findings show that it is an effective and efficient approach for sending real-time secure digital images

across insecure networks [17]. The paper also proposes a method for encrypting images quickly. With a procedure, the processes of diffusion and confusion are merged into one phase. A CST (chaotic shift transform) is used to modify an image's pixel locations, and the replacements of column and row are utilized to jumble the values of pixel at the same time. The outcomes of the simulations and analyses show that this technique is secure, has low complexity of the time complexity, and could withstand chosen-plaintext, known-plaintext attacks and brute force [18].

3. An Attractor of Aizawa

An attractor of Aizawa is a framework of equations which can be applied to 3D coordinates, develops in such a manner that the outcomes coordinate map into a 3D form. In this instance, a sphere within a tube-like structure pierce one of its axes. The Aizawa attractor is shown in equation (1). The actual equations are quite straightforward [19]:

$$\left.\begin{aligned} dx &= x*(z-b) - y*d \\ dy &= x*d + y*(z-b) \\ dz &= c + a*z - z^3/3 - x^2 + z * f * x^3 \end{aligned}\right\} \quad (1)$$

In which a, b, c, d, z, and f are constant values in a specific range. Each previous coordinate is insert to equations, the outcome value multiplied via a value of time chosen, and then added to the previous. Then, added to the prior value, x, y, and z are variables in the equation should have initial values as shown in Figure 1.

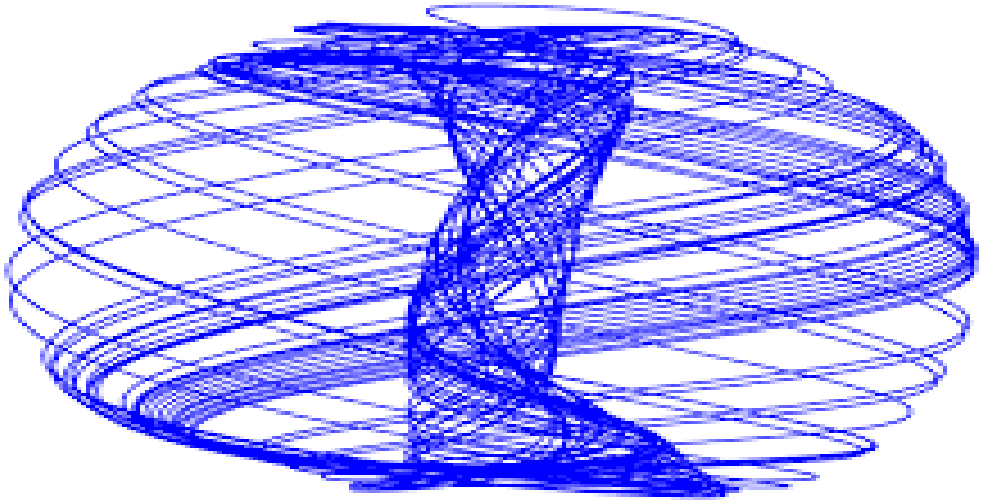


Figure 1: Aizawa Attractor [19]

4. Rabbit- Lightweight Encryption Algorithm

The Rabbit algorithm aims to increase speed and security. The rabbit was designed to operate more quickly compared to conventional ciphers. There are no cryptographic issues at this time [20]. It might produce a 128-bit pseudo-random bit output block on each cycle utilizing a combination of internal 513 bits and 64-bit IV as input and a 128-bit secret key. One carry bit, eight 32-bit counter variables, and eight 32-bit state variables make up the internal bits. As many as 2^{64} blocks of cipher output cannot be distinguished from random generator's output by an attacker without the key, necessitating the computation of 2^{128} key combinations. Block diagrams for the rabbit algorithm are shown in Figure 2.

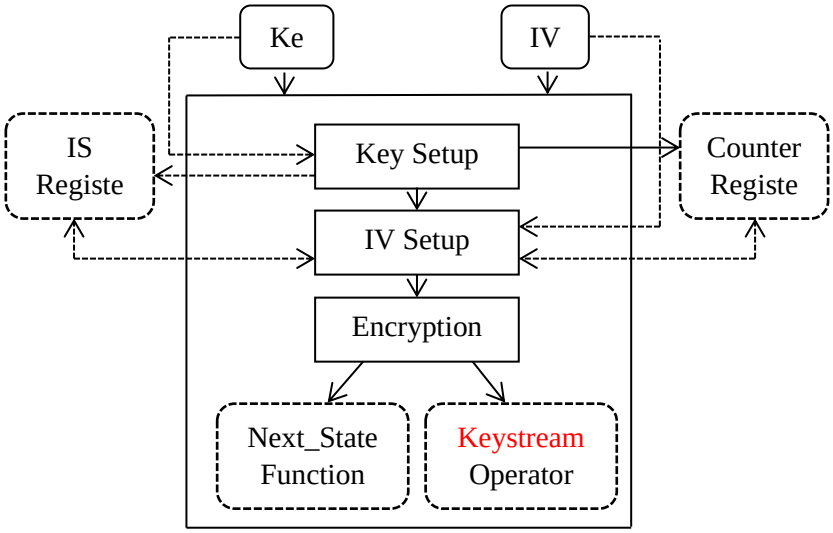


Figure-2 A block diagram of Rabbit algorithm

4.1 *Key Set-up Scheme*

The algorithm's initial step is to set up a key. There are eight 16-bit subkeys that make up 128-bit key. State and counter variables are then created using sub keys. The key and initial state variable values, $X_{j,o}$, and initial counters, $C_{j,o}$, have a one-to-one relationship with the eight counters and eight state variables. Eight sub-keys make up K [127::0]: k0 = K [15...0], k1 = K [31::16],... k7 = K [127::112]. As seen in Eq.2 and Eq.3, state and counter variables are initialized by using sub keys. [20].

$$x_{j,0} = \begin{cases} k_{[j+1\ modulo\ 8]} \diamond k_j & for\ j\ even \\ k_{[j+5\ modulo\ 8]} \diamond k_{[j+4\ mod\ 8]} & for\ j\ odd \end{cases} \quad (2)$$

And

$$c_{j,0} = \begin{cases} k_{[j+4\ modulo\ 8]} \diamond k_{[j+5\ mod\ 8]} & for\ j\ even \\ k_j \diamond k_{[j+1\ modulo\ 8]} & for\ j\ odd \end{cases} \quad (3)$$

To decrease the connection between internal state variables and key bits, the system is after that iterated for 4 times in accordance with the next state function. Counter variables are then re-initialized with the use of the procedure in Eq. 4 after that [21]:

$$c_{j,4} = c_{j,4} \oplus x_{(j+4\ mod\ 8),4} \quad (4)$$

Inverting the counter system for every j in Eq.4 to prevent the key from being recovered.

4.2. *IV Setup Scheme*

A duplicate of this master state should be altered in accordance with IV scheme, and the internal state should be the internal state that adheres to the key setup scheme. The counter state is altered by IV setup technique in response to IV. To accomplish this, 256 bits of counter state are all XORed with the 64-bit IV. IV [63::0] stands for the IV's 64 bits. The counters were modified to [21]:

$$\left.\begin{aligned} c_{0,4} &= c_{0,4} \oplus \mathrm{IV}^{[31::0]} \\ c_{2,4} &= c_{2,4} \oplus \mathrm{IV}^{[63::32]} \\ c_{4,4} &= c_{4,4} \oplus \mathrm{IV}^{[31::0]} \\ c_{6,4} &= c_{6,4} \oplus \mathrm{IV}^{[63::32]} \end{aligned}\right\} \quad \left.\begin{aligned} c_{1,4} &= c_{1,4} \oplus (\mathrm{IV}^{[63::48]} \diamond (\mathrm{IV}^{[31::16]}) \\ c_{3,4} &= c_{3,4} \oplus (\mathrm{IV}^{[47::32]} \diamond (\mathrm{IV}^{[15::0]}) \\ c_{5,4} &= c_{5,4} \oplus (\mathrm{IV}^{[63::48]} \diamond (\mathrm{IV}^{[31::16]}) \\ c_{7,4} &= c_{7,4} \oplus (\mathrm{IV}^{[47::32]} \diamond (\mathrm{IV}^{[15::0]}) \end{aligned}\right\} \quad (5)$$

The system is after that iterated for 4 iterations in order to make sure that each state bit depends nonlinearly on each IV bit. All 2^{64} IVs produce a unique key stream thanks to the IV's alteration on the counter.

4.3 *Next-state Function*

The iteration of system that had been specified through the next equations [21] is at the heart of the Rabbit algorithm:

$$x_{0,i+1} = g_{0,i} + (g_{7,i} <<< 16) + (g_{6,i} <<< 16)$$
$$x_{1,i+1} = g_{1,i} + (g_{0,i} <<< 8) + g_{7,i}$$
$$x_{2,i+1} = g_{2,i} + (g_{1,i} <<< 16) + (g_{0,i} <<< 16)$$
$$x_{3,i+1} = g_{3,i} + (g_{2,i} <<< 8) + g_{1,i}$$
$$x_{4,i+1} = g_{4,i} + (g_{3,i} <<< 16) + (g_{2,i} <<< 16)$$
$$x_{5,i+1} = g_{5,i} + (g_{4,i} <<< 8) + g_{3,i}$$
$$x_{6,i+1} = g_{6,i} + (g_{5,i} <<< 16) + (g_{4,i} <<< 16)$$
$$x_{7,i+1} = g_{7,i} + (g_{6,i} <<< 8) + g_{5,i}$$
$$g_{j,i} = ((x_{j,i} + c_{j,i+1})^2 \oplus ((x_{j,i} + c_{j,i+1})^2 \gg 32)) mod\ 2^{32}$$

All of the addition operations are modulo 2^{32} in this case. Figure 3 illustrates this linked system that represents the equations above as a graph with different shifting values depending on the location. The counters are increased before each repetition, as stated below.

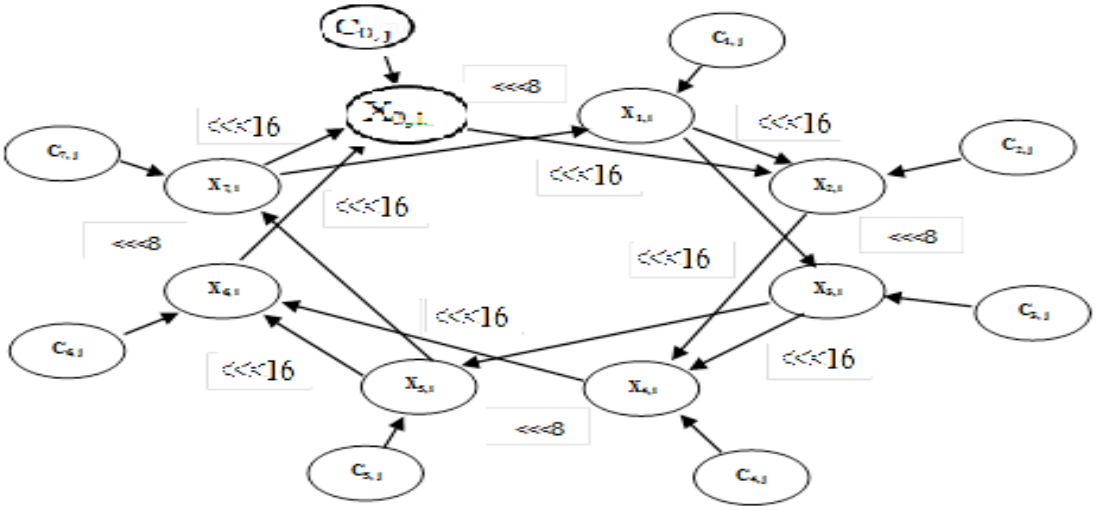


Figure 3: Graphical system

4.4 *Counter System*

The counters' dynamics are described as follows:

$$c_{0,i+1} = c_{0,i} + a_0 + \emptyset_{7,i} \quad mod\ 2^{32}$$
$$c_{1,i+1} = c_{1,i} + a_1 + \emptyset_{0,i+1} \quad mod\ 2^{32}$$
$$c_{2,i+1} = c_{2,i} + a_2 + \emptyset_{1,i+1} \quad mod\ 2^{32}$$
$$c_{3,i+1} = c_{3,i} + a_3 + \emptyset_{2,i+1} \quad mod\ 2^{32}$$
$$c_{4,i+1} = c_{4,i} + a_4 + \emptyset_{3,i+1} \quad mod\ 2^{32}$$
$$c_{5,i+1} = c_{5,i} + a_5 + \emptyset_{4,i+1} \quad mod\ 2^{32}$$
$$c_{6,i+1} = c_{6,i} + a_6 + \emptyset_{5,i+1} \quad mod\ 2^{32}$$
$$c_{7,i+1} = c_{7,i} + a_7 + \emptyset_{6,i+1} \quad mod\ 2^{32}$$

in which the carry bit for the counter, $\emptyset_{j;\,i+1}$, is provided by

$$\emptyset_{j,i+1} = \begin{cases} 1 & if\, c_{0,i} + a_o + \emptyset_{7,i} \geq 2^{32} \wedge j = 0 \\ 1 & if\, c_{j,i} + a_j + \emptyset_{j-1,i+1} \geq 2^{32} \wedge j > 0 \\ 0 & Otherwise \end{cases}$$

In addition, a_j constants are specified as follows:

$$a_o = 0X4D34D34D \qquad a_1 = 0XD34D34D3$$
$$a_2 = 0X4D34D34 \qquad a_3 = 0X4D34D34D$$
$$a_4 = 0XD34D34D3 \qquad a_5 = 0X34D34D34$$
$$a_6 = 0X4D34D34D \qquad a_7 = 0XD34D34D3$$

4.*5 Extraction Scheme*

The extraction scheme is the last stage, in which the XOR operation is performed on various state registers to produce a total of eight 16-bit key stream registers. In addition, the plain text bit stream is then XORed with these key bits.

$$s_i^{[15..0]} = x_{0,i}^{[15..0]} \oplus x_{5,i}^{[31..16]} \qquad s_i^{[31..16]} = x_{0,i}^{[31..16]} \oplus x_{3,i}^{[15..0]}$$
$$s_i^{[47..32]} = x_{2,i}^{[15..0]} \oplus x_{7,i}^{[31..16]} \qquad s_i^{[63..48]} = x_{2,i}^{[31..16]} \oplus x_{5,i}^{[15..0]}$$
$$s_i^{[79..64]} = x_{4,i}^{[15..0]} \oplus x_{1,i}^{[31..16]} \qquad s_i^{[95..80]} = x_{4,i}^{[31..16]} \oplus x_{7,i}^{[15..0]}$$
$$s_i^{[111..96]} = x_{6,i}^{[15..0]} \oplus x_{3,i}^{[31..16]} \qquad s_i^{[127..112]} = x_{6,i}^{[31..16]} \oplus x_{1,i}^{[15..0]}$$

In which s_i represent 128-bit key stream block at i^{th} iteration. To encrypt/decrypt, extracted bits are XORed with plain-text/cipher text.

$$c_i = p_i \oplus s_i$$
$$p_i = c_i \oplus s_i$$

The i^{th} 128-bit plaintext and ciphertext blocks are denoted by p_i and c_i respectively.

5. Proposed Method

The proposed method mixes the Rabbit algorithm, a lightweight encryption algorithm, with a new key generation method based on the Aizawa Attractor. Several steps used to encrypt the images are shown in Figure 4.

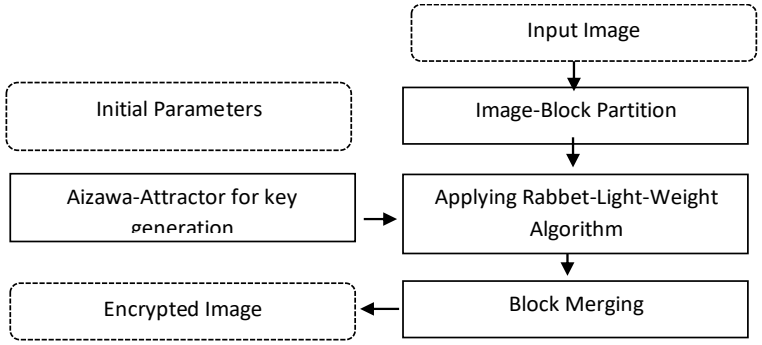


Figure 4: Flowchart of the Proposed Encryption Method

Figure 5 splitting color image

As it has been depicted in Figure 5, the color image is initially divided into the 3 color bands of red, green, and blue. These bands will merge into one array and then partition into blocks of a specific size for encryption. Padding zeros are applied to the last block if it is smaller than the other blocks.

The encryption method follows the following steps:
1. Input a color image.
2. Split the image into three color bands.
3. Convert the color bands into vectors.
4. Merge the three vectors into one vector.
5. Segment the vector into specific-sized blocks of bits.
6. Generate a three-dimensional key using the Aizawa Attractor.
7. Use one of the three key dimensions for the blocks' permutation.
8. Use the other two key dimensions for scheduling the block key.
9. Concatenate the blocks into a vector.
10. Split the vector above into three vectors.
11. Reshape each vector into a two-dimensional array of the same size as the input.
12. Reconstruct the three arrays into one three-dimension array that represents the encrypted image.

Table 1: key generation samples

Generated numbers	Removing floating point	Hexadecimal
0.65574069915659	65574069915659	'98AD310214'
0.03571167857419	03571167857419	'085094F86E'
0.84912930586878	84912930586878	'C5B40CF70C'
0.93399324775755	93399324775755	'D97655F40D'
0.67873515485777	67873515485777	'9E07C48EA9'
0.75774013057833	75774013057833	'B06CD4D112'
0.74313246812492	74313246812492	'AD0625939C'
0.39222701953417	39222701953417	'5B528D5B0E'
0.65547789017756	65547789017756	'989D86DC81'
0.17118668781156	17118668781156	'27DB858F43'

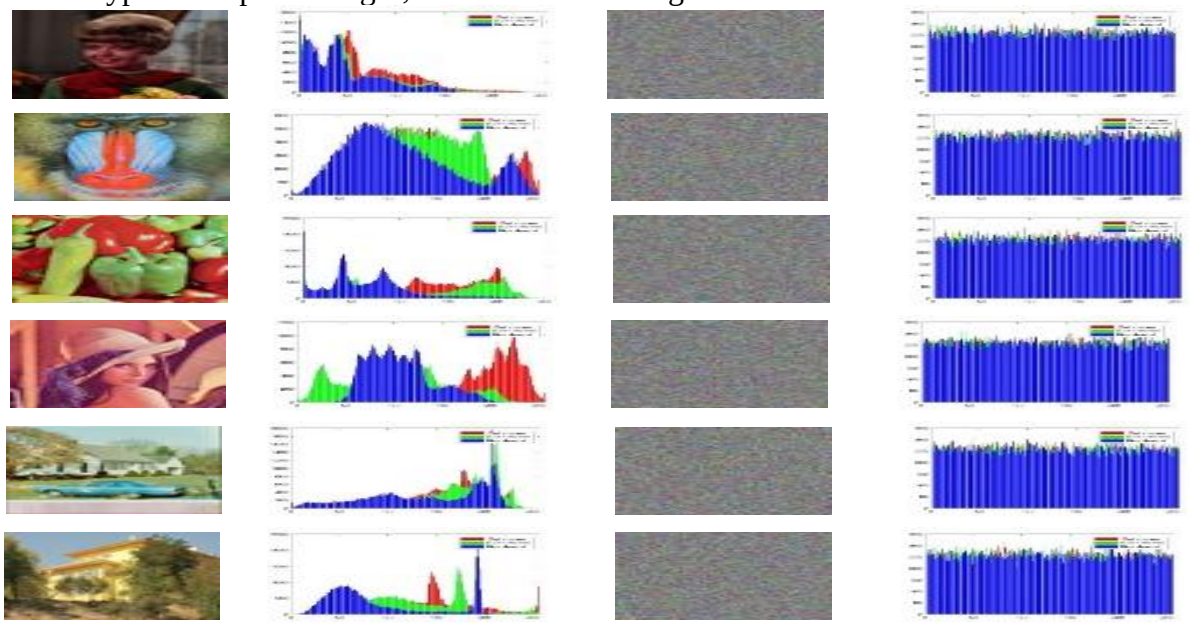
The key generation step employs the Aizawa Attractor to generate a sequence of real numbers using equations 1, 2 and 3, as explained in Table 1. These numbers are processed by removing the floating point and obtaining several specific digits after the floating point. These numbers are then converted to hexadecimal and merged into one sequence to form a three-dimensional key for the encryption algorithm. The blocks are not selected for encryption in sequential order, but rather according to one dimension of the generated key.

The decryption method follows the steps below:
1. Input encryption image.
2. Split the encrypted image into three color bands.
3. Convert the color bands into vectors.
4. Merge the three vectors into one vector.
5. Segment the vector into same-sized blocks of bits for encryption.

6. Generate the same key used in encryption using the Aizawa Attractor under the same initial conditions.
7. Use one of the three key dimensions to reorder the blocks.
8. Use the two other dimensions for the block key schedules for decryption.
9. Concatenate the blocks into a vector.
10. Split the above vector into three vectors.
11. Reshape each vector into a two-dimensional array of the same size as the input.
12. Reconstruct the three arrays into one three-dimensional array that represents the decrypted image.

6. Experimental Results

The proposed method is tested on a set of standard images, namely, images of a woman, a baboon, a pepper, Lena, a car, and a house. The histogram test is run as the initial test on both the encrypted and plain images, as can be seen in Figure 6.



a) **b)** **c)** **d)**

Figure6 a) plain-images, b) Histograms of the plain images, c) Cipher-images, d) Histogram of the cipher images

Figure 6 shows the differences between the histograms regarding plain images and resulting images, which are seemingly uniform in shape. A uniform histogram means that no values are more important than other values, i.e., there are no correlations between the resulting values. The second test is the correlation test as illustrated in Eq.6, which is applied to each of the three-color bands (red, green, and blue) of an input image and an output image (encrypted image). In particular, horizontal correlations are used to test pixels with row neighbours, vertical correlations are used to test pixels against column neighbours, and diagonal correlations test pixels with diagonal neighbours, as shown in Figure 7.

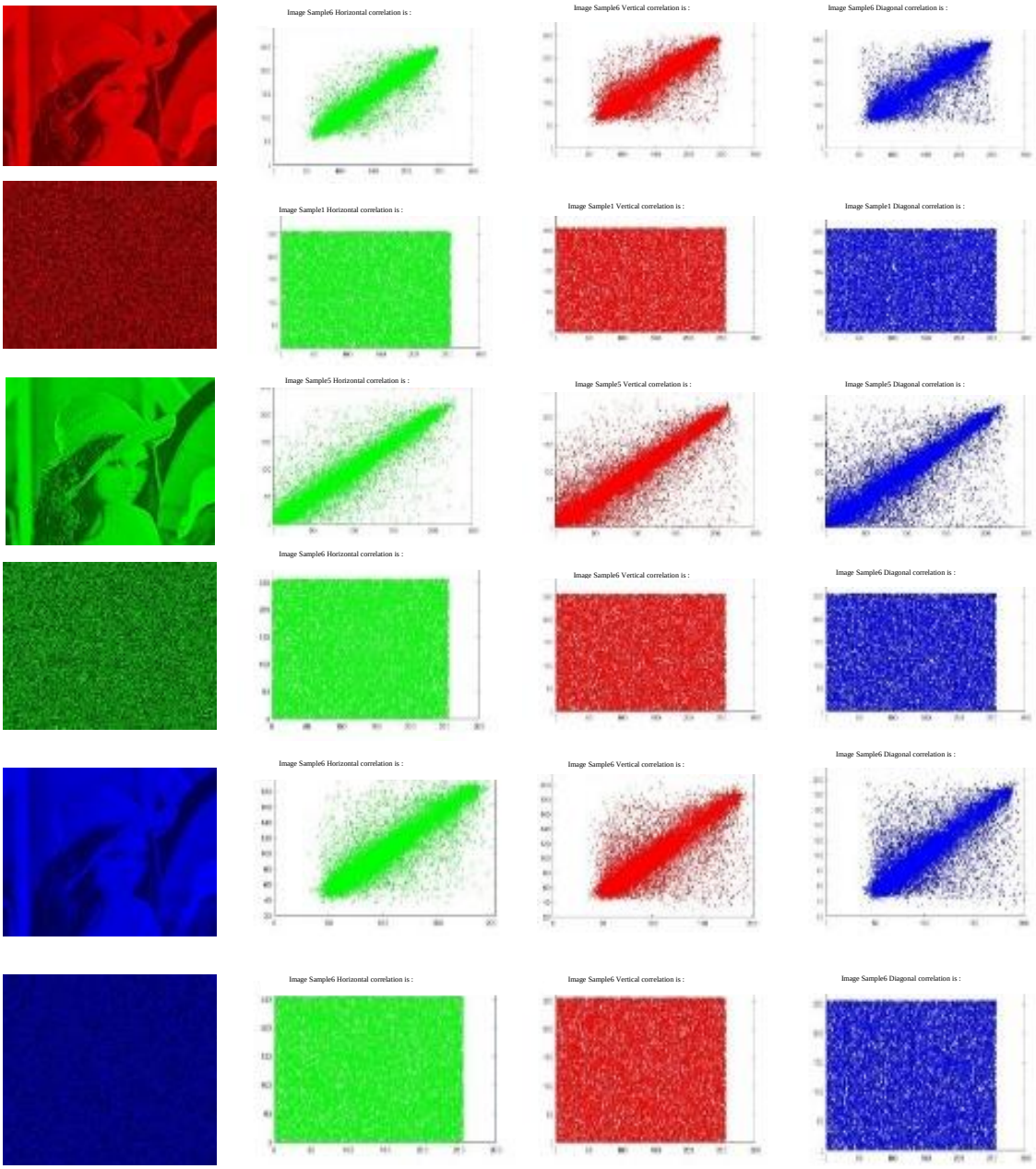


Figure 7: Correlation Test for

$$r = \frac{\sum(x_i - \dot{X})(y_i - \grave{Y})}{\sum(x_i - \dot{X})^2 \sum(y_i - \grave{Y})^2} \qquad (6)$$

r=correlation coefficient
x_i=values of the x-variable in an image
$\dot{X}$=mean of the values of x-variables
Y_i= values of the y-variable in an image
$\grave{Y}$= mean of the values of y-variables
Figure 8: shows the results of the diagonal, horizontal and vertical, and correlations in entered images as well as after A, B, and C encryption in, respectively.

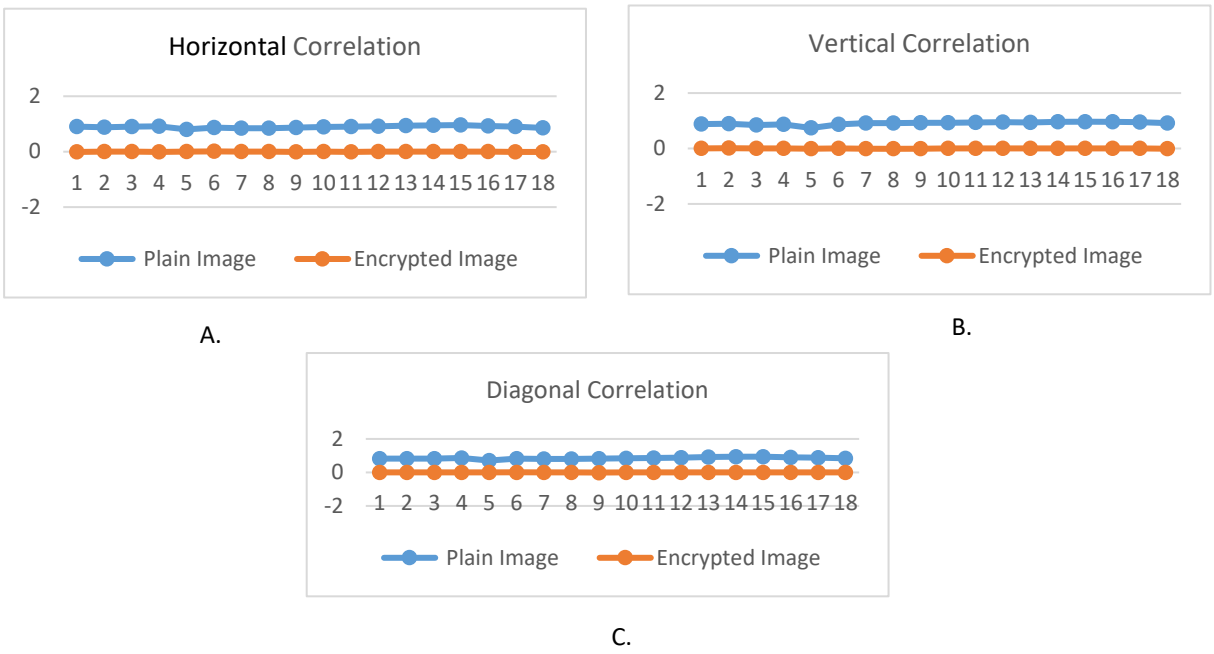


Figure 8 : Correlation test result for all images before and after encryption
A. Horizontal B. Vertical C. Diagonal Correlations

Following Aizawa Attractor generates a key, it will be saved after the floating-point is eliminated and its digits are converted to hexadecimal values in order to be utilized in validation steps of an encryption process. Given that the keyspace must be larger than 2^{128}, brute force attack risk is evaluated using keyspace analysis. The Aizawa Attractor's initial parameters, x_o, y_o, z_o, a, b, c, d, and f, each with precision 1010, are used in this study to find the keyspace. Thus, it could be shown that the keyspace is vast and the encryption method might withstand brute force attacks because the keyspace is computed as $(1010)^8$. As a result, it is possible to rely on this approach to generate keys that are more secure. The time consumed in applying the proposed algorithm is shown in Table 2 for the encryptions (Enc.) and decryptions (Dec.) of images with sizes, namely, 128×128, 256×256, and 512×512 bits.

Table 2: Time-consuming for first hybrid encryption/decryption algorithm

	128x128		**256x256**		**512x512**	
Image size	Enc. Time (Ms)	Dec. time (Ms)	Enc. time (Ms)	Dec. time (Ms)	Enc. time (Ms)	Dec. time (Ms)
1	0.001800	0.008000	0.004900	0.002000	0.016500	0.005200
2	0.001400	0.004000	0.005500	0.002400	0.012600	0.005200
3	0.001200	0.002000	0.005100	0.001700	0.011100	0.006000
4	0.001300	0.003000	0.006000	0.001900	0.018200	0.004200
5	0.001200	0.002000	0.004600	0.002000	0.012800	0.003800
6	0.001300	0.005000	0.005100	0.002500	0.015500	0.003300
Average	0.001800	0.008000	0.004900	0.002000	0.016500	0.005200

Table 3 displays the results of the NIST tests, which were used to confirm that a proposed key has high security and can withstand a variety of attacks. An algorithm of a proposed framework passed all these tests because of the modifications in each experimentally known test key (the test key size was 65,536 bits).

Table 3: Tests of NIST for a proposed Algorithm

#	Tests	P-Value	Status
1	Run Testing	0.101245	Pass
2	Serial Testing	0.027221	Pass
3	random excursion variant testing	0.690011	Pass
4	random excursion testing	0.766554	Pass
5	Non-overlapping template matching testing	0.890005	Pass
6	Frequency Monobit Testing	0.003333	Pass
7	Maurer's universal statistical testing	0.000041	Pass
8	the longest run of 1s in a block testing	0.001111	Pass
9	Linear complexity Testing	0.023333	Pass
10	Frequency test within the Block testing	0.034423	Pass
11	Discrete Fourier Transform testing	0.788651	Pass
12	Cumulative sums Testing	0.004567	Pass

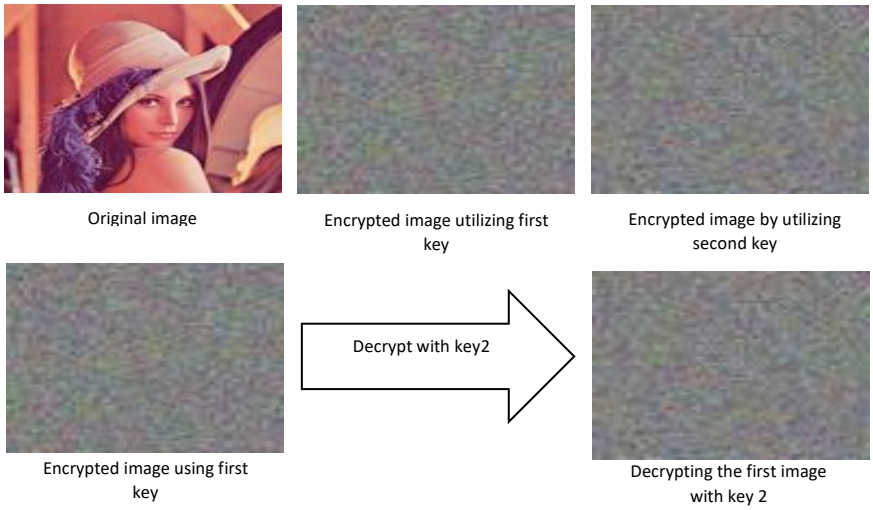


Figure 9: Lena image encryption with two keys has a small change in the initial

The Key Sensitivity analysis determines whether the secret key is sensitive to the best method for image encryption. That is, a secret key must produce an entirely different image with a minor change. With a slight modification to the initial condition, Figure 8 illustrates the encryption of the Lena image with the use of a different key.

There are several objective tests for determining the similarities between two images, but the image quality test can be used to find the dissimilarities between them. The image quality test for encrypted images compares that encrypted image with the original image. The MSE must be low between the encrypted and original images. Other tests depend on MSE, like SNR and PSNR tests, both of which look for small MSE values. The SSIM is utilized to investigate the interior correlations of image objects. Also, the entropy test is applied to the encrypted images, and the values for all images are near the maximum required bits (8 bits). Table 4 explains these tests in detail for a sample of images.

Table 4: Image quality Test

#-No.	MSE	PSNR	SNR	SIM	Entropy
1	8760.508374	0.00454	2.01572	117.324	7.98129
2	8046.581023	0.00512	1.78946	103.971	7.72136
3	9885.706659	0.00444	1.54534	151.293	7.96683
4	9308.746932	0.00466	1.87381	114.079	7.71967
5	8283.911598	0.00485	1.87665	128.197	7.74252
6	9617.87948	0.00485	1.60606	108.261	7.61548
Average	8983.889011	0.00474	1.78451	120.521	7.79119

7. Conclusion

This work presents an efficient method for image encryption that combines a lightweight method (the Rabbit Algorithm) with a three-dimensional dynamic chaotic map (the Aizawa Attractor). To encrypt each block, multiple key stream sequences are produced by the hybrid method, as controlled by the initial conditions. The combination provides an efficient, lightweight encryption algorithm that supports image transmission in modern applications, such as the IoT. The experiments show that the generated key streams pass the NIST test for randomness. Moreover, the quality of the resulting image was tested. The results demonstrate that there is no relation between the encrypted and input images. Furthermore, several tests, including the MSE, PSNR, SSIM, and entropy test, were performed to investigate the quality of the proposed method, revealing promising results. Finally, the time needed for encryption and decryption was considered. The Aizawa image encryption technique can survive several forms of attacks while preserving confidentiality or security because there are no downsides or restrictions. The suggested approach can be used in subsequent research to tackle a variety of additional multimedia formats, including text data, databases, and videos. The suggested approach might also be utilized to generate keys. Lastly, it might be combined simply with another lightweight encryption method.

References

[1] E. Tariq and E. Falih,"Image Encryption and decryption using CAST-128 with proposed adaptive key," *J. Coll. Educ.*, no. 5, pp. 89-100, 2019.

[2] M. Salih, R. Abdulaali and N. Falih, "A proposed lightweight image encryption using ChaCha with hyperchaotic maps," *Period Eng. Nat. Sci. (PEN)*, vol. 8, no.4, pp. 2138- 2145, 2020.

[3] A. Noori and E. Falih, "Hiding the Type of Skin Texture in Mice based on Fuzzy Clustering," *J. Bag. Sci.*, vol. 17, no.3, pp. 967-972, 2020.Doi: 10.21123/bsj.2020.17.3(Suppl.).0967.

[4] M. S. Mahdi, R. A. Azeez and N. F. Hassan,"A suggested super salsa stream cipher," *Iraq. J. Comp. Inform.*, vol.44, no.2,pp. 5-10, 2018.

[5] L. JIAO, Y. HAO and D. FENG, "Stream cipher designs: a review," *Sci. China Inform. Sci.*, vol.36, pp.1-25,2020, doi.org/10.1007/s11432-018-9929-x.

[6] S. Raizada, "Some results on analysis and implementation of HC-128 stream cipher," PhD, Applied Statistics Unit, *Indian Stat. Inst.*, West Bengal, India Kolkata, pp.1-147, 2015.

[7] G. Bansod, "RAGHAV: A new low power SP network encryption design for resource-constrained environment," Cryptology ePrint Archive,2021.

[8] J. R. Naif, G. H. Abdul-Majeed, and A. K. Farhan, "Secure IOT system based on chaos-modified lightweight AES," *Int. Conf. Adv. Sci. Eng. (ICOASE)*. IEEE, 2019.

[9] S. Sharma,T. Kumer, R. Dhaundiyal, A. K. Mishra, N. Duklan, "Improved method for image security based on chaotic-shuffle and chaotic-diffusion algorithms," *Int. J. Elec. Comp. Eng.*, vol. 9, no.1, 2019.

[10] M. H. Shaheen, "Hybrid Encryption Algorithms Over Wireless Communication Channels," CRC Press, pp.1-268, eBook ISBN9781003051428,2021. Available: doi.org/ 10.1201/ 97810 03051428.

[11] S. S. Tadkal and M. V. Mahalinga, "Secure Transmission of Data using Rabbit Algorithm," *IRJET*, vol. 4, no. 5, pp. 3079-3083, 2017.

[12] B. Ghanbari and J.F. Gómez-Aguilar, "Two efficient numerical schemes for simulating dynamical systems and capturing chaotic behaviors with Mittag–Leffler memory," *Eng. with Comp.*, pp.1-29, 2020.

[13] A. T. Sadiq, A. K. Farhan and S. Hassan, "A proposal to improve RC4 algorithm based on hybrid chaotic maps," *J. Adv. Comp. Sci. Technol. Res*, vol.6, no.4, pp.74-81, 2016.

[14] W. Wang, "Reversible data hiding scheme based on significant-bit-difference expansion," *IET image processing*, vol. 11,no.11, pp.1002-1014,2017.

[15] H. Wen, C. Zhang and L. Huang, "Security Analysis of a Color Image Encryption Algorithm Using a Fractional-Order Chaos," Entropy, vol. 23, no.2,2021. Available: https: //doi. org/ 10.3390 / e23020258.

[16] A. N. Mazher, J. Waleed and A. T. MaoLood, "Developed Lightweight Cryptographic Algorithms for the Application of Image Encryption: A Review," *J. Al-Qadisiy. Comp. Sci. Math.*,vol. 13, no.2 pp.1-11, 2021.

[17] A. Akgul, B. Gürevin and I. Pehlivan, "Development of micro-computer based mobile random number generator with an engineering application," *Integration the VLSI J.*,vol.18,no.2, pp.1-16, 2021.Available: doi.org/10.1016/ j.vlsi.2021.04.010.

[18] X.Y. Wang, and L. Z. M. Li, "A color image encryption algorithm based on Hopfield chaotic neural network," *Opt. and Laser Eng.*, vol. 115, pp. 107-118, 2019. Available: https://doi.org/ 10.1016 / j.optlaseng.2018.11.010.

[19] L. You, E. Yang & G. Wang, "A novel parallel image encryption algorithm based on hybrid chaotic maps with OpenCL implementation," *Soft Comp.*, vol. 24, no. 16, pp.12413–12427, 2020. Available: https://doi.org/10.1007/s00500-020-04683-4.

[20] Z. Man, J. Li, X. Di and O, Bai, "An image segmentation encryption algorithm based on a hybrid chaotic system," *IEEE Access*, vol. 7, pp. 103047-103058, 2019.

[21] Q. Xu, K.Sun,C. Cao and C. Zhu, "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Opt. and Lasers Eng.*, vol. 121, pp.203-214, 2019.