IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 9, September, 2025, pp. 3203-3215
Journal homepage: http://iieta.org/journals/mmep

Chaos-Based Pseudo Random Generator Function for Medical Image Encryption

Baydaa H. Helal*, Nibras Z. Salih, Noor A. Yousif, Ashwaq T. Hashim

Control and Systems Engineering Department, University of Technology, Baghdad 10001, Iraq

Corresponding Author Email: Baydaa.H.Hilel@uotechnology.edu.iq

https://doi.org/10.18280/mmep.120924

Received: 4 January 2025
Revised: 3 May 2025
Accepted: 14 May 2025
Available online: 30 September 2025

Keywords:
pseudo-random generator, medical images, Chebyshev polynomials, image encryption, RC6 algorithm, chaotic systems

ABSTRACT

Image data can be considered vital components in medical information systems. More data will be available for research as medical image analysis techniques and imaging technology develop. Therefore, it is vital to safeguard those images. In multimedia applications, image encryption techniques play a crucial role in ensuring the security and legitimacy of digital images. This study proposes a new random generator to encrypt medical images. An efficient 128-bit random number generator with a suitable random sequence is needed to enhance the security associated with the suggested encryption method. Using the RC6 algorithm and Chebyshev polynomials, this system encrypts the medical images based on the generated sequence. The proposed pseudorandom bit generating uses the RC6 encryption technique as a function, where the parameters are (k, l), and the start values are (x_0, y_0) of Chebyshev polynomials maps are used as subkeys for RC6 algorithm. The output of the random generator is combined with initial vector (IV) in chain mode to start the random generator. The IV is initialized with the output of the MD5 hash function for the full image. According to the entropy and optimal values of the analysis parameters, the standard image "Lena" and simulation results on different medical images confirm a high success rate and an arbitrary pattern of the pixels. Therefore, all sequences produced by the proposed random generator successfully pass the National Institute of Standards and Technology (NIST) randomness tests. The proposed method offers fast implementation and high efficiency, and resilience against various attacks, while providing strong security features.

1. INTRODUCTION

Medical centers have recently increased the use of online medical image scanning. Given the importance of medical image data, encrypting this data would inevitably prevent its access and distribution [1, 2]. Also, the line between medical data and the Internet has become more open to the outside world, making it vulnerable to many hacking attempts. Hence, security development has focused on protecting medical information due to the previously unheard-of safety risks related to the protection of patients, including reports, photos, and other data [3]. Medical image encryption technology uses common encryption systems, such as RC6, AES, Blowfish, etc. block ciphers. The reliability of block ciphers is determined by the key generation mechanism and the algorithm used in the encryption process, which depends on the key length. In data encryption techniques, random chaos is the numbers that are not related to each other or any other number in the sequence to provide a potential artificial random generator [4, 5]. The features of number chaos are connected to the constraints of encryption systems. Several techniques are used in chaotic structures, including summation, sensitivity to change in the initial context, and identification of variables. The layout of random generators is essential in encryption applications [6]. Obtaining identical random numbers at the sending point and destination is a difficult problem to solve. Therefore, a program that produces a series of numbers independently is known as a pseudo-random generator (CSG) that produces an extended random pattern using a short random number as input. CSG can have a repeated process, which is a one-way process. The CSG's characteristics include a chaotic system for performing random change and responding to an initial value [7, 8].

This paper introduces the following contributions:

1) Proposed chaos depends on the pseudo-bit random generating. The RC6 encryption algorithm is employed as a function in the suggested pseudorandom bit generating, where the parameters are (k, l), and the start values are (x_0, y_0) of the Chebyshev polynomials maps are used as sub-keys for the RC6 algorithm.
2) For RC6 block ciphers, the connection between input and output blocks is completely arbitrary. This design must be one-to-one, in which case each input block is mapped to a distinct output block. Consequently, image encryption relies mainly and mostly on randomness; the randomness rate impacts the strength and robustness of any security systems and increases their complexity.
3) The proposed CSG method is further enhanced by Chebyshev polynomial maps encryption that increases privacy. A Chebyshev chaotic mapping is a technique for

effective security implementation.

4) A suggested technique is used in the encryption process to encrypt the pixels with a random sequence related to the input medical picture bits.
5) Due to our keyspace being suitably wide, our design encourages resilience to brute-force searching attacks.

This study is structured as follows: Section 2 presents related works, and the preliminary plan is provided in Section 3. Section 4 outlines the general steps of the CSG process. Section 5 presents the results of the experiment and comparisons. Our efforts come to an end in Section 6.

2. RELATED WORKS

Block cipher is designed for constant blocks and key sizes that use two crucial processes, which are diffusion and confusion. A popular technique of block cipher is based on chaos theory. The CSG is usually generated by chaotic maps and used as encryption keys. This section provides a summary of the most significant methods for encrypting chaotic pictures.

In 2017, Saraswathi et al. [9] presented symmetric stream encryption technology for encrypting medical images. Statistical analyses were executed using secret streams created for cipher streams. The findings demonstrated the suitability of the suggested encryption method for real implementations.

In 2018, Öztürk and Kılıç [10] performed four separate time analyses of Chebyshev structures using a single parameter to generate pseudorandom numbers. The authors combined the different attractions of the converted systems which was the final attraction. Also, Shakiba, [11] applied one-dimensional Chebyshev mappings to encrypt various chaotic pictures.

Randomization is applied to the image elements using a breadth-first technique to build a chaotic process of distribution matrices. The authors provided strong security against noise and information corruption scenarios. In 2020, Tutueva et al. [12] provided a method for developing chaotic cryptosystems with a high parameter space to use chaotic maps with adaptive symmetry. The authors compared CSG which depended on standard Zaslavsky maps, and classical Zaslavsky maps using multiple parameters for instability examination.

In 2021, Krishnamoorthi et al. [13] utilized a turbulent logistic technique to generate arbitrary values in the chaotic structure. A final sequence was produced using a two-dimensional logistic technique as the base map. The unpredictability of the base map was improved using Duffing's map as a perturbation map.

In 2021, Louzzani et al. [14] offered a generative execution of Chebyshev polynomial functions to specific control parameter values. The generated function displays chaotic results to secure pictures versus different types of attacks which were demonstrated by the bifurcation graph and Lyapunov expression.

Sreedharan and Eswaran [15] utilized the Advanced encryption standard depending on Chebyshev's chaotic behavior to eliminate some iterations of the shuffling process. The proposed model can reduce the computing complexity and transmission overhead through an upgraded simpler algorithm to encrypt conversation signal features throughout communication. In 2022, Dridi et al. [16] presented a cartographic system based on block cipher to evaluate the system's effectiveness, this system involves a pseudorandom generator of chaotic patterns. The author suggested a cyclic replacement process to carry out an effective chaos. A specific conjunction matrix connects the four distinct chaotic maps that comprise the pseudorandom generator.

In 2024, Rashidi [17] developed CSG for the blocked cipher that was utilized to encrypt digital communication by (CMOS) complementary metal-oxide semiconductor circuitry. Arbitrary keys and data encryption in the unitary network were incorporated into the technique proposed by the author. The original key was employed for creating a fresh dynamic key which was used for encryption.

3. PRELIMINARIES

3.1 Chebyshev polynomials and their properties

One type of nonlinear dynamic process that depends on the starting parameters is called a chaotic process. Random structures can produce highly unpredictable chaos [18]. There are many types of chaotic processes, including Henon mapping, logistic mapping, and Chebyshev polynomial mappings.

So, chaotic elements are arranged to reorder the original features and extract the permuted characteristics. The source of chaotic sequences is Chebyshev polynomial maps, which operate on the image pixel values using a nonlinear process to determine the setting parameters. Each permutation key is generated by sorting chaotic numbers in increasing or decreasing order and is defined in the following equation [19]:

$$x(n+1) = T_K(x_n) = (k \times arc\cos(x_n)) \quad (1)$$

$$y(m+1) = T_K(y_m) = (l \times arc\cos(y_m)) \quad (2)$$

where, $(x_n, y_m) \in [-1, 1]$ and $(k, l) \in (2, \infty)$ are the control parameters.

3.2 RC6 encryption algorithm

A robust cipher should be resistant to every type of encryption analysis and system-breaking assaults for a reliable indicator of its effectiveness. RC6 stands as one of the top block cipher algorithms that were a modification of the RC5 method [20].

Algorithm 1: RC6 Encryption

Input: Let (r) is the number of rounds for the key $S[0 \ldots, 2r+3]$, and the plaintext is kept in four w-bit input registers (A, B, C, and D).

Output: Cipher text stored in **(A, B, C, D)**

$B = B + S[0]$
$D = D + S[1]$
for $i = 1$ tor do
{
$t = \mathrm{B} * (2B + 1) \ll \log(w)$
$\mathrm{u} = \mathrm{D} * (2D + 1) \ll \log(w)$
$A = ((A \oplus t) \lll \mathrm{u}) + S[2i]$
$C = ((C \oplus u) \lll \mathrm{t}) + S[2i + 1]$
$(\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}) = (\mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{A})$
}
$A = A + S[2r + 2]$
$C = C + S[2r + 3]$

Algorithm 2: RC6 Decryption

Input: Let (r) is the number of rounds for key S[0…,2r + 3], and cipher text in four w-bit input registers (A, B, C, and D).

Output: Plaintext stored in **(A, B, C, D)**

$C = C - S[2r + 3]$

$A = A - S[2r + 2]$

for $i = r$ down to 1 do

{

$(A, B, C, D) = (D, A, B, C)$

$u = D * (2D + 1) \ll \log(w)$

$t = B * (2B + 1) \ll \log(w)$

$C = ((C - S[2i + 1]) >>> t) \oplus u$

$A = ((A - S[2i]) >>> u) \oplus t$

}

$D = D - S[1]$

$B = B - S[0]$

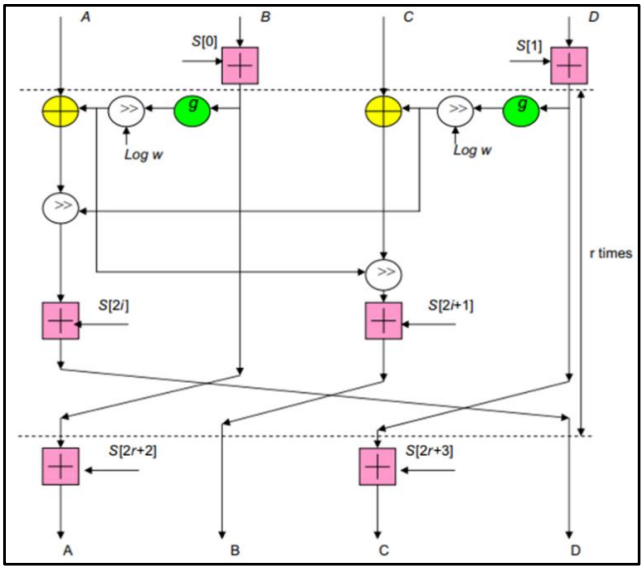


Figure 1. RC6 encryption algorithm scheme

Three primary parameters of RC6 are w, r, and b, where w represents the word block size, r describes the encryption rounds, and b is the user-specified key length. Algorithm 1 and Algorithm 2 use four 32-bit registers that can handle 128-bit data transfer blocks and are compatible with key sizes that range between (128–2040) bits in length. RC6 has a key space large enough to prevent brute attacks [21]. Figure 1 illustrates the encryption process in the RC6 technique.

4. THE ENCRYPTION ALGORITHM PROPOSED

Many different algorithms for encryption especially for digital images are introduced to preserve the need for powerful applications of image encryption. Several ideal properties have made chaos-based encryption algorithms increasingly popular, including unexpectedly strong sensitivity for beginning conditions and other settings. To enhance the reliability of the suggested encryption technique, a pseudorandom bit generator based on Chebyshev polynomials and RC6 is utilized for generating random sequences. Then it is used to encode medical images using the chaotic shift keying principle. Algorithm 3 shows the steps of the proposed encoding technique. This algorithm for decryption is explained in Algorithm 4. Figure 2 describes the main structure of the proposed algorithm.

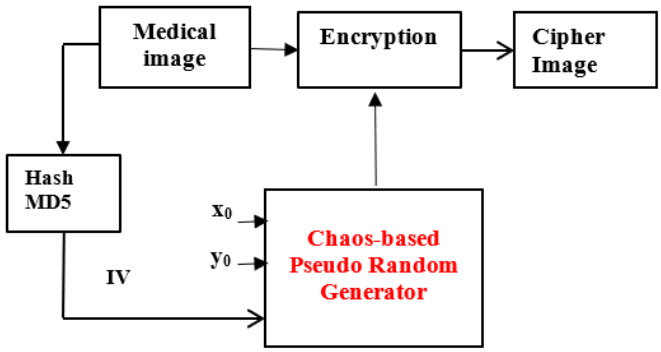


Figure 2. The proposed algorithm

Algorithm 3: Image Encryption Steps

Input: *MI* // Medical Image
(*R, C*) // width and height of *MI*

Output: *MEnc* // Encrypted Medical Image.

1. Convert *MI* into a vector in one dimension *MI'* = {*MI'*(*1*), *MI'*(*2*),..., *MI'*(*l*)}, where *l* = *R* × *C*.

2. Convert vector *MI'* to binary to generate *BMI'*

3. Generate chaotic sequence *S* using the proposed pseudo-random bit generator using the algorithm (5).

4. Code the vector *MI'* by embedding it in the generated random sequence *S*.

for i = 1 to (W × H)
if *BMI'*(*i*) = 1
ME(*i*) = S(*i*)
else
ME(*i*) = -D1(*i*)
endif
endfor

5. Convert the real sequences *ME* to binary using the following formula:

MEbin=Round (*ME*+0.5)

Where the integer numbers larger than (0) became (1) and the integer numbers less than (0) became (0).

6. Convert binary coded sequence to decimal then store in *MEnc*.

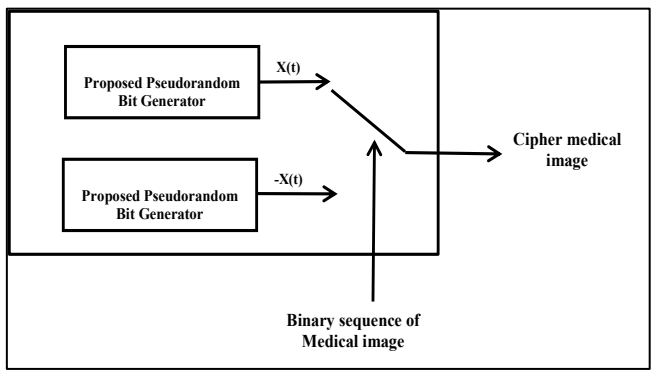


Figure 3. The encryption process block design

Chaos signals are employed during the encryption procedure to transfer a binary order of medical pictures. One signal of chaos $x(t)$ at a time is sent to encode a digital symbol. For instance, if a binary signal of bit "1" at (t) time is to be sent. Also, $x_1(t)$ is transferred, when $x_0(t)$ binary signal of bit "0" (I.e., $x_0(t) = -x_1(t)$) can be shared. Two distinct chaotic

systems or in the same system via distinct features that can generate chaotic signals $x_1(t)$ and x_2(t). The main structure for the encoding technique is displayed in Figure 3. Signal modulation is performed using the quadratic chaotic mapping, which is expressed as the following:

$$s(t) = \begin{cases} x_1 & \text{symbol 1 is transmited} \\ x_0 & \text{symbol 0 is transmited} \end{cases}$$

The receiver generates the same random sequence *S* using the identical start conditions as the transmitter, finally a deciphering main structure is described in Figure 4.

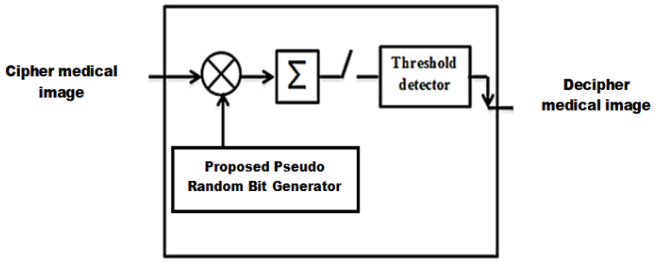


Figure 4. Main structure of the deciphering process

Algorithm 4: Image Decryption Steps

Input: *MEnc* // medical image encrypted
R, C // Width and Height of *MEnc*
Output: *MI'*
1. Convert *MEnc* to binary to generate *MEbin.*
2. Generate chaotic sequence *S* using the proposed pseudo-random bit generator using an algorithm (5).
3. Perform the following equation:
G = Round ((*MEbin*(I) - 0.5)
4. Multiply the G by the chaotic sequence *S.*
H = *S* × G
if (H(*i*) > T)// The threshold value is T, (i.e., T = 0)
BMI'(*i*) = 1
else
BMI'(*i*) = 0
5. Converted *BMI'* to decimal to generate *MI'*.

4.1 Chaos-based pseudo random generator

A binary pseudorandom sequence is produced by preprocessing the real values for two Chebyshev polynomials x and y and then merging them using the RC6 method. The proposed chaos-based pseudo random generator depends on two Chebyshev polynomials that are described in Eqs. (1) and (2).

The suggested technique uses the round functions of the RC6 encryption to provide a 128-bit random number generator that achieves a sufficient random sequence and an enhanced generation rate. The RC6 algorithm uses the starting values (x_0, y_0) and the parameters (k, l) of the Chebyshev polynomial maps as subkeys. To begin the random generator, the output is combined with the initial vector (IV) in a series mode.

The generator may be used to produce a password for the signing process or to produce a key for any security system. There are numerous applications for this generator, including games and the creation of random patterns. The IV is the hash code for the source medical image that is utilized as the first vector of the generator, and Figure 5 shows the comprehensive design chaos-based pseudo random generator.

The seeds, or keys, are the sets x_0 and y_0. Therefore, the rounds of RC6 random block cipher technique are the foundation of the suggested chaos-based pseudo random generator. The block-chaining encryption mode combination of two instances of RC6 is used to create chaining of RC6 (CRC6). Each CRC6 block is started with a primary key, represented by the x_0 and y_0, respectively; these keys serve as the pseudorandom generator's seed and the input of the first RC6 is set to the general IV.

According to the Algorithm 5, this system produces long random that are 128 bits. The first RC6 input is (x_0), the second RC6 input is (y_0), and m_i is the intermediate, as shown in Figure 6. Then, S_i is the CRC6 output.

Algorithm 5. Chaos-based Pseudo Random Generator

Input:
x_0, y_0, k, l // initial conditions and parameters for Chebyshev Polynomial
MI // Medical image
R, C // Medical image dimensions
Output: *SBin* // Binary Random Sequence
Step 1: Compute the hash code of medical image *MI* such as follows:
IV = MD5 (*MI*)
Step 2: Let l = (R×C)/16
Step 3: Apply Chebyshev map Eqs. (1) and (2) using four initial values x_0, y_0, k, and l to generate two subkeys x_0 and y_0. Then transform the x_0 and y_0 into 128-bit unsigned integer such as following:
- Set the four initial values x_0, y_0, k, l as secret values.
- Call Eqs. (1) and (2) using four initial values x_0, y_0, k, l to generate x_0 and y_0.
- Transform x_0 and y_0 into 128-bit unsigned integer such as below:

$$x_{normalizes} = \frac{x_0+1}{2}$$
$$x_0 = round(x_{normalizes} \times (2^{128} - 1)$$
$$y_{normalizes} = \frac{y_0+1}{2}$$
$$y_0 = round(y_{normalizes} \times (2^{128} - 1)$$

Step 4: Generate random number sequence *S* based on Chebyshev map such as follows:
For i = 0 to l
m_i = RC6 (IV, x_i) // where x_i is 128-bit used to generate 44 subkeys for first RC6.
S_i = RC6 (m_i, y_i) // where y_i is 128-bit used to generate 44 subkeys for second RC6.
$x_{i+1} = x_i \oplus m_i$
$y_{i+1} = y_i \oplus m_i$
$S_i = y_{i+1}$
$IV = IV \oplus y_{i+1}$
$SBin_{ij}$ = Convert S_i to binary j = 1, …, 8
Endfor

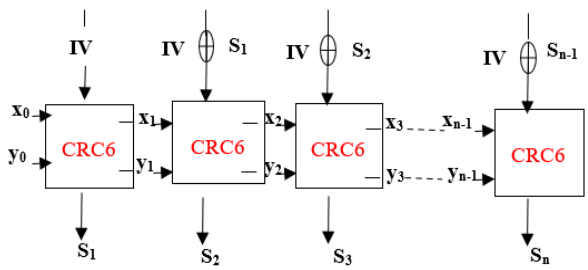


Figure 5. The main structure of proposed chaos-based pseudo random generator

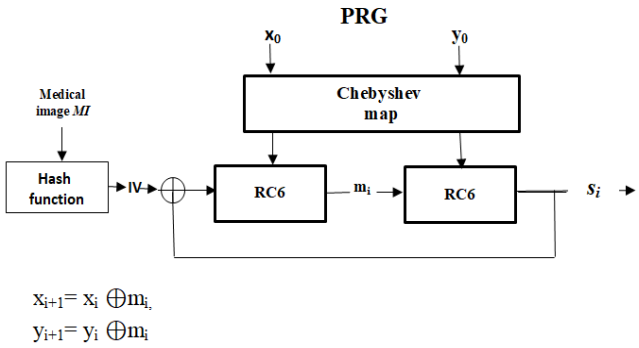


Figure 6. Block diagram of proposed the CRC6

5. PERFORMANCE ANALYSIS

In this paper we used image samples from different medical image modalities, as shown in Figure 7, and the input parameters and initial values for Chebyshev maps x_0, y_0, k, l are set to 0.398, 0.456, 0.784, 0.982 successively. The primary parameters of RC6 are set as the number of round r = 20, b = 128-bits and w = 32-bits The following section examines the parameters of the performance model to evaluate the proposed algorithms.

5.1 Histogram analysis

The encoding outcomes of the source and the cipher images along with the histogram are displayed in Figure 8. Histogram assessments produce the individual pixel value intensities of the histogram for the cipher and source pictures.

So, a graph that shows the number of pixels in a given image at each varying intensity value is called a histogram. The comparison is made between the source pictures and the visual representations of the related cipher pictures. This comparison showed that the histograms of the cipher pictures are almost identical, while the histograms of the source pictures are completely different for every picture.

Moreover, a histogram is a representation that describes the total number of pixels at different intensity levels, using the axes to show the standard distribution of picture pixels. The histogram of the real image typically contains some unevenly distributed features [22].

Thus, an effective encryption technique requires stable distribution in the encrypted image. And to evaluate the success of the encryption technique, the distribution histogram shows the time numbers for the intensity values that appear in the pictures [23].

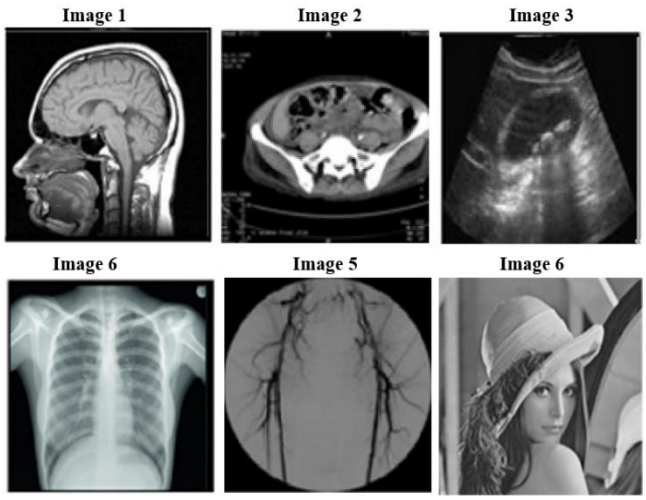
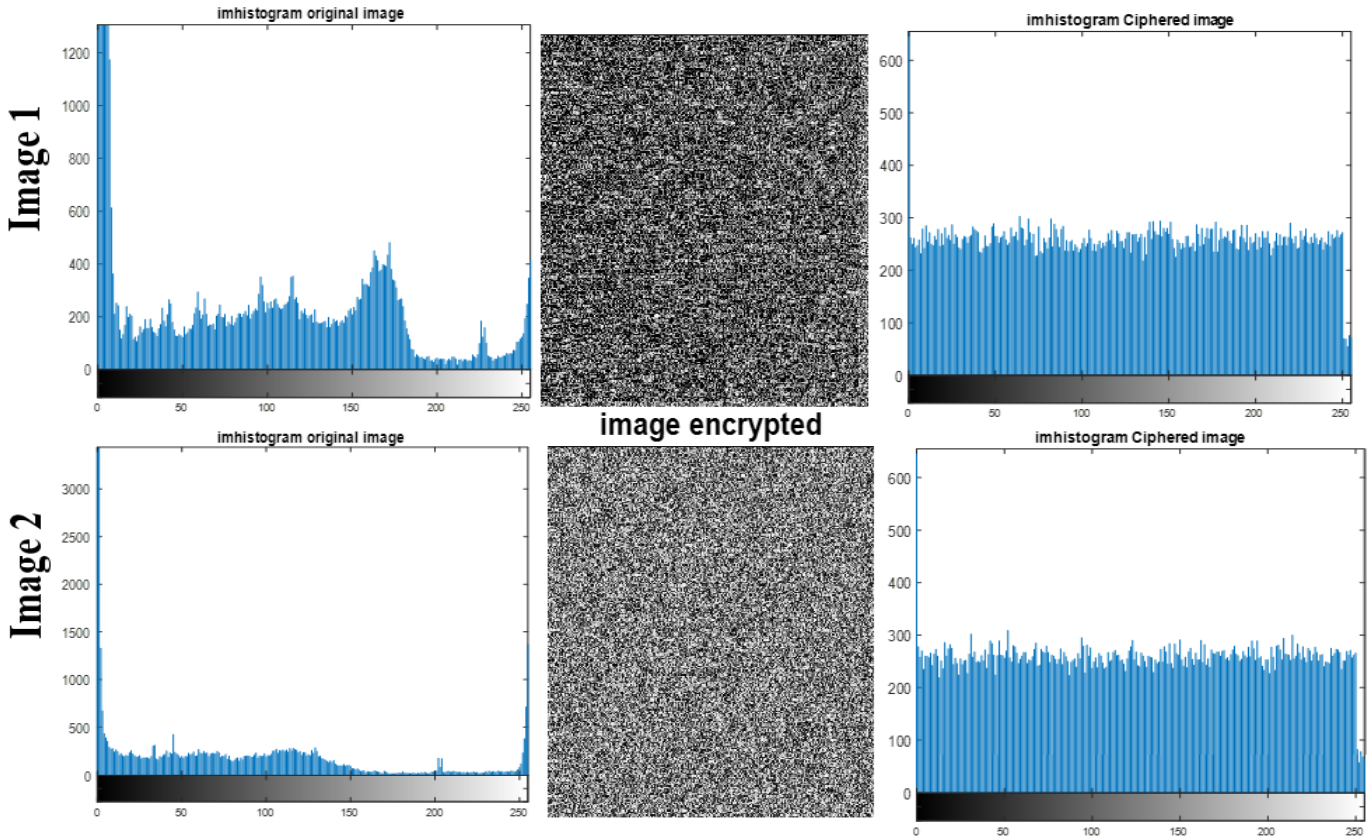


Figure 7. Test images



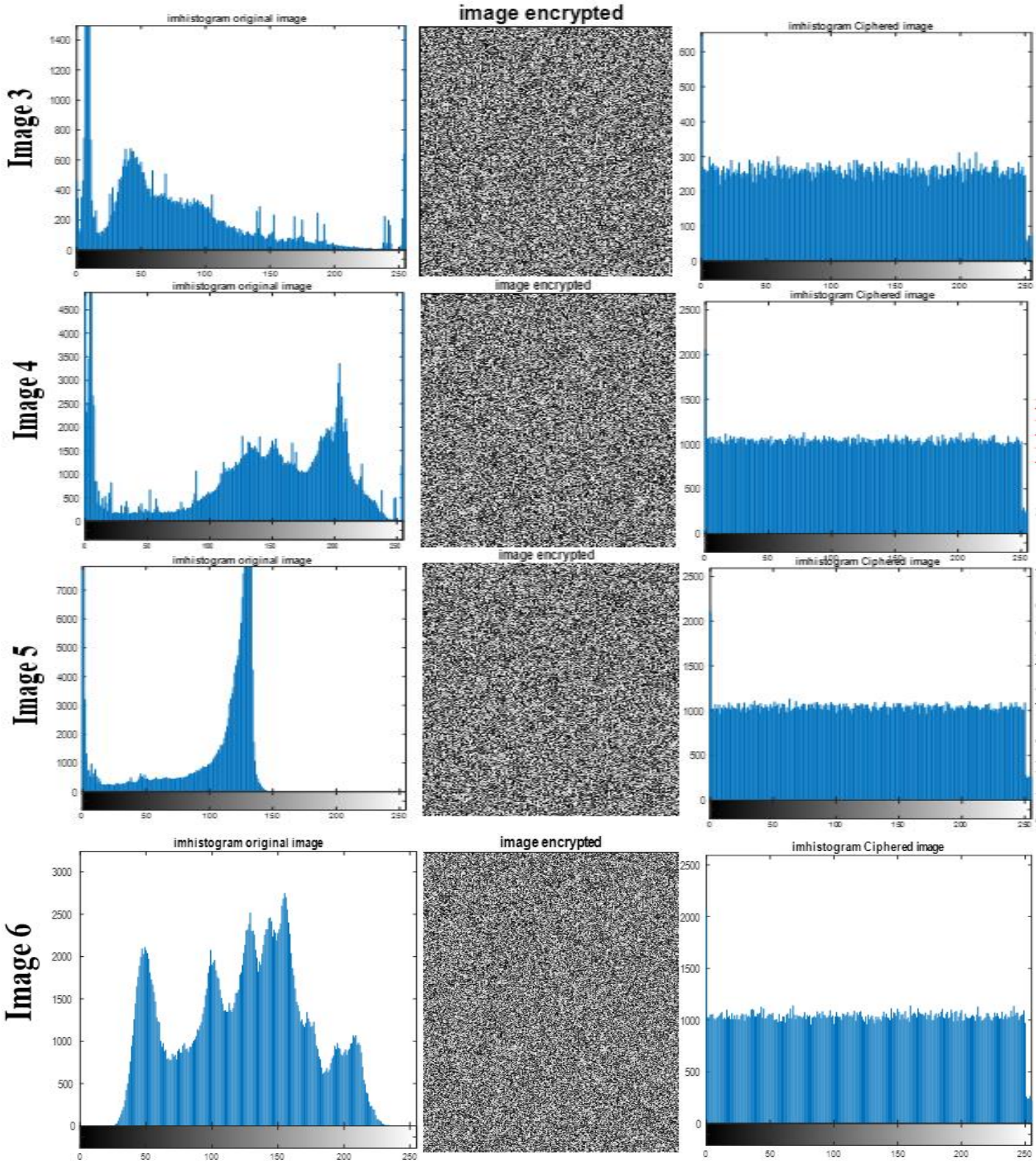


Figure 8. The suggested decryption and encryption mechanism for medical pictures: Medical picture histogram, Picture encrypted, Encrypted picture histogram

5.2 Correlation coefficients analysis

Correlation Coefficient (C.C) refers to the connection between two factors. When Image and data encryption are closely related it usually has one correlation. So that, the encrypted and source pictures have the same appearance., it means that the encryption process was not able to hide the actual image information.

Whereas the actual image and its encryption are completely different has zero correlation. The encrypted image loses some features and is very different from the actual image. When the C.C is -1, it indicates that the encrypted picture is on the opposite side of the real picture [24].

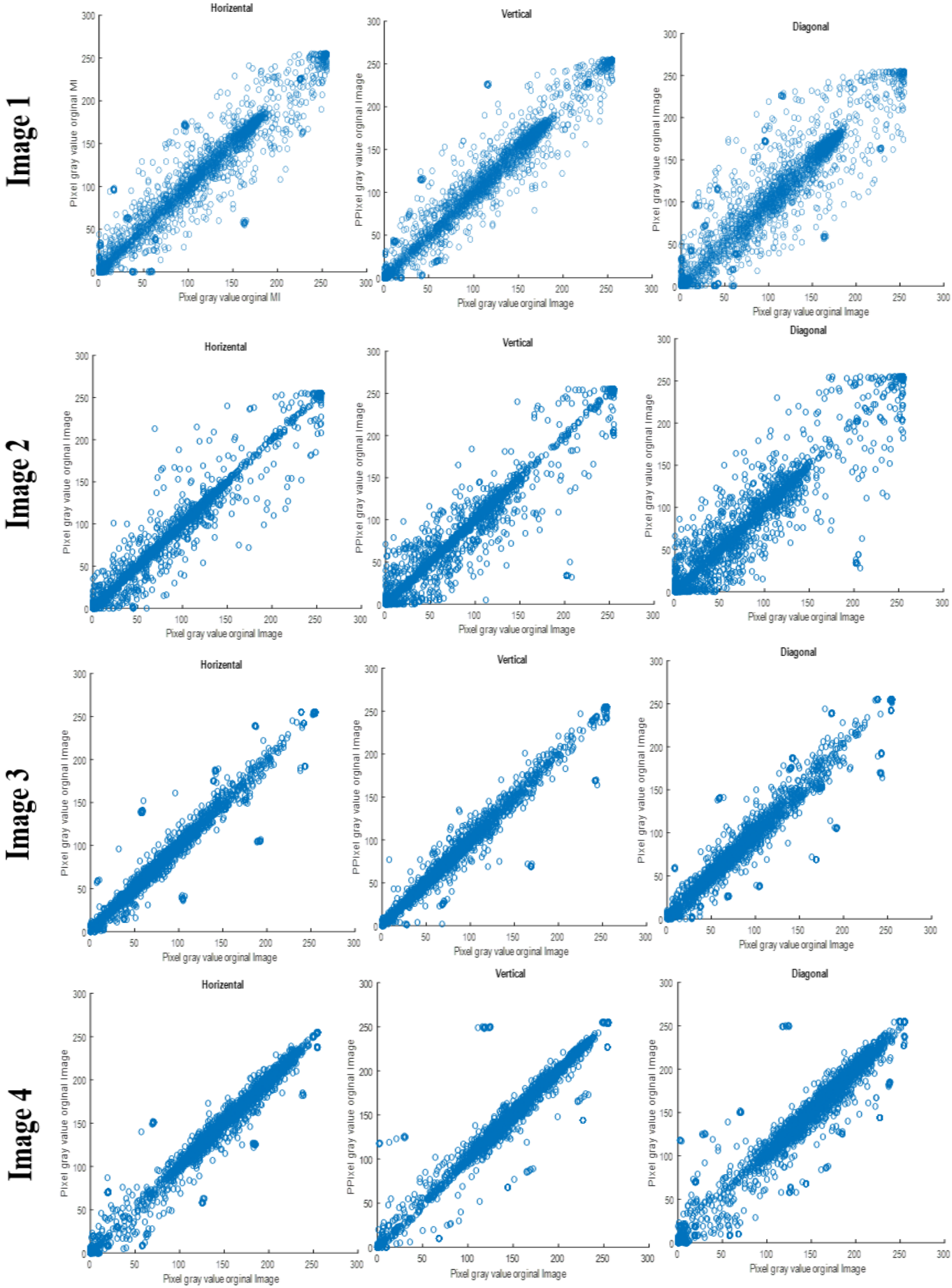


Image 1
Image 2
Image 3
Image 4
Horizental
Vertical
Diagonal
Pixel gray value orginal MI
PPixel gray value orginal Image
Pixel gray value orginal Image

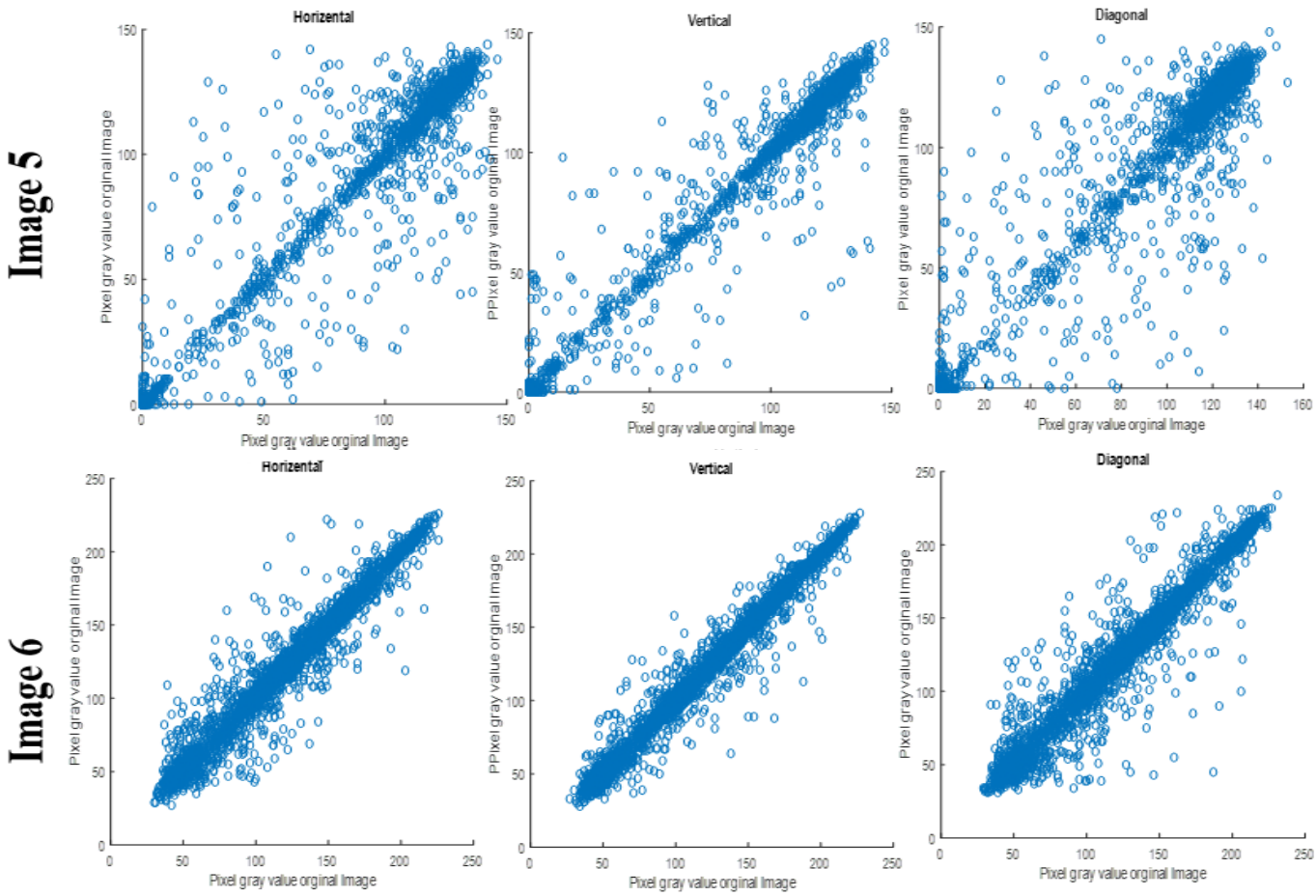
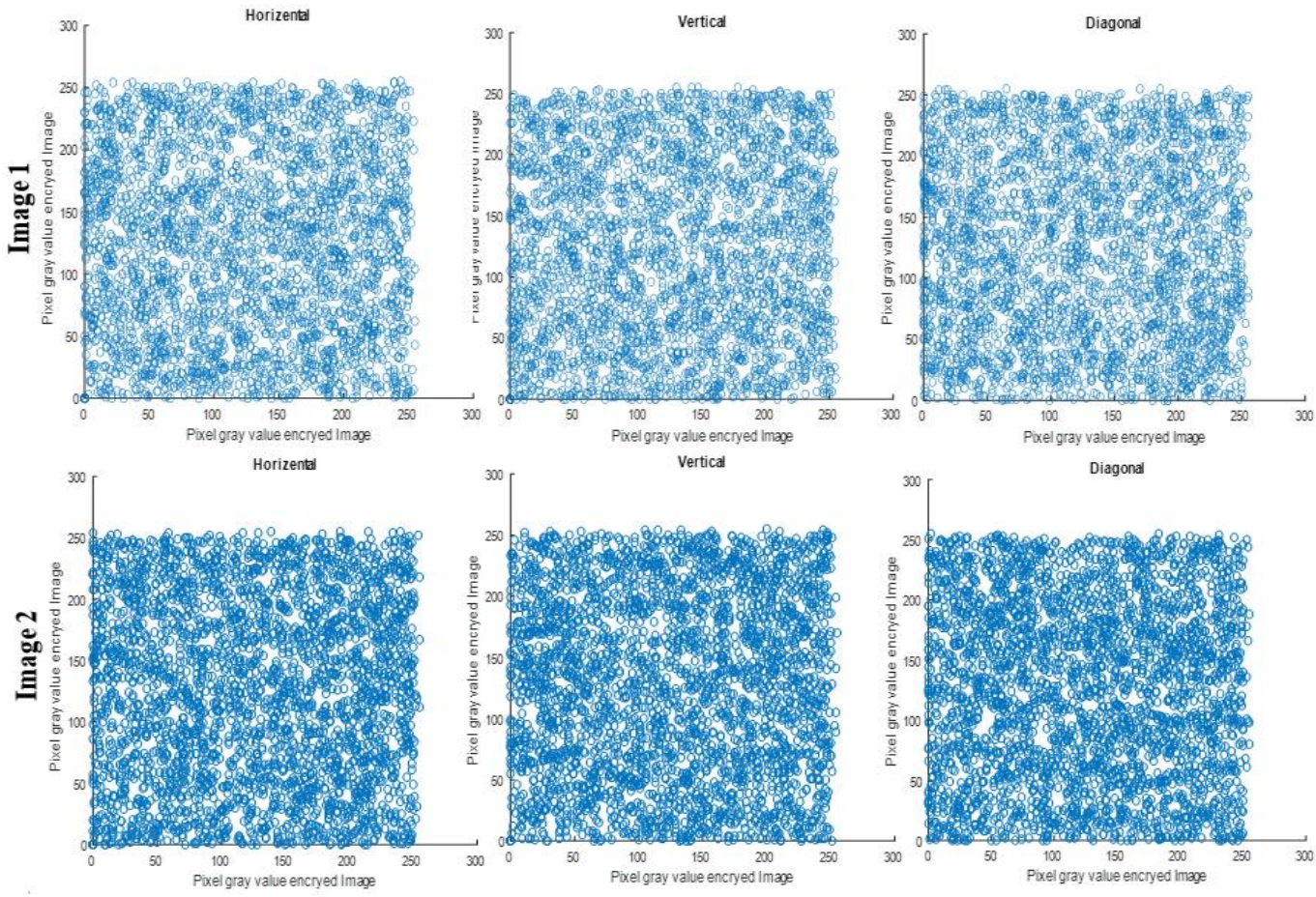


Figure 9. Correlation coefficients for the source medical pictures



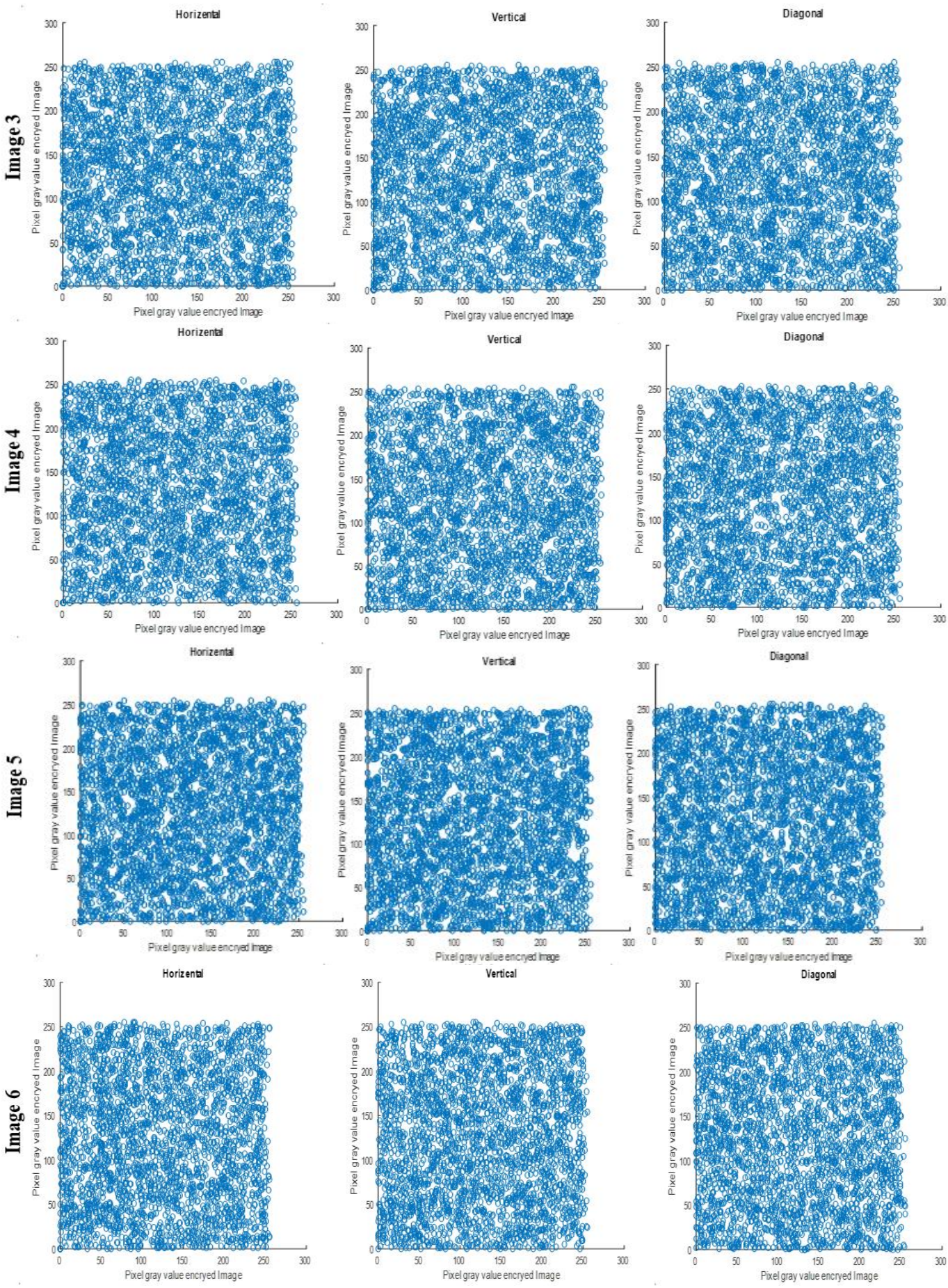


Figure 10. Correlation coefficients for the encrypted pictures

Therefore, successful encryption leads to lower C.C. values, and are calculated using Eq. (3) by ρ_{xy}. Figure 9 shows the (C.C) of the source picture for the Horizontal Correlations (H.C), Vertical Correlations (V.C), and Diagonal Correlations (D.C) and Figure 10 explains the (C.C) of the encrypted picture. In Table 1, the (C.C) of the proposed approach is presented in five pictures.

$$\rho_{xy} = \frac{\text{con}\,(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \tag{3}$$

$$\text{con}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\left([x_i - M(x)][y_i - M(y)]\right)$$

$$M(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$M(y) = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - M(x)\right)^2$$

$$V(y) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - M(y)\right)^2$$

where, $\text{con}(x,y)$ represents the covariance between x and y, ($V(x)$ and $V(y)$) represent the variance of the parameters x and y, $M(x)$ and $M(y)$ describe the mean of two parameters, x and y.

Parameters x and y, $M(x)$ and $M(y)$ describe the mean of two parameters, x and y.

Table 1. C.C for the proposed technique

Pictures	H.C	V.C	D.C
Pic1	-0.00104	-0.00163	-0.00144
Pic2	-0.00132	-0.00332	-0.00133
Pic3	0.00013	- 0.00055	- 0.00011
Pic4	-0.00146	-0.00133	-0.00246
Pic5	-0.00908	-0.00997	-0.00707

5.3 Analyzing the entropy

Entropy refers to the mean amount of bits in the grayscale levels of the picture, which correspond to its data value. So, a rise in entropy indicates greater ambiguity regarding the picture's contents. To analyze an arbitrary performance for the encrypted picture, entropy H(C) for the discrete two-dimensional images is determined using the entropy statistics specified as:

$$H(C) = \sum_{i=0}^{N_c-1} P(c_i) \times \log_2 \frac{1}{P(c_i)} \tag{4}$$

$H(C)$ represents the encrypted image's entropy, whereas $P(c_i)$ describes the probability of occurrence number for every level where (i = 0–255). At the probability rate $P(c_i)$ is 2^{-8}, when the entropy is highest, it will be equal to eight. So, the Images with more ambiguous details and minimal leakage have higher entropy. Data entropy for any image indicates that it cannot be predicted by the probability of different gray levels occurring. Therefore, the image with different gray levels represents the highest amount of unpredictability, which shows its degree of secrecy. The details of the entropy for the proposed technique are mentioned in Table 2 which shows that the entropy of the encryption picture is close to eight bits, which indicates that the proposed approach is the best [25]. In Table 2, a comparison of the entropy to both source pictures and the proposed technique is presented below.

Table 2. Entropy comparison

Pictures	Source Pictures	Proposed Technique
Pic1	7.0957	7.9975
Pic2	5.7758	7.9969
Pic3	6.9069	7.9975
pic4	7.2566	7.9995
pic5	5.7201	7.9887

5.4 Definition of UACI and NPCR

Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) were initially reported according to Mao and Chen [26, 27].

The NPCR and UACI metrics are usually used to evaluate the picture security for the encryption techniques against different attackers. Greater NPCR and UACI scores indicate significant protection from distinct attacks. Let T^1 and T^2 represent the images of the ciphertext in plaintext before and after changing by one pixel. Eq. (5) defines a bipolar array A and the value of each pixel across the grid in T^1 and T^2. Eqs. (6) and (7) can mathematically define the NPCR and UACI respectively described below.

$$A(i,j) = \begin{cases} 0, \text{if } T^1(i,j) = T^2(i,j) \\ 1, \text{if } T^1(i,j) \neq T^2(i,j) \end{cases} \tag{5}$$

$$\text{NPCR}(T^1,T^2) = \sum_{i,j} \frac{A(i,j)}{D} \times 100\% \tag{6}$$

$$\text{UACI}(T^1,T^2) = \sum_{i,j} \frac{|T^1(i,j) - T^2(i,j)|}{R \cdot D} \times 100 \tag{7}$$

where, the symbol (D) denotes the total of the pixel numbers in the ciphertext, whereas the symbol R represents the maximum pixel value. NPCR focuses on the absolute pixel number that changes during different attackers. As well as UACI focuses on the mean distinction among pair ciphertext images.

Table 3. Comparison of the NPCR and UACI metrics

Pictures	The Proposed Technique	
	NPCR	UACI
Pic1	99.6093	33.4835
Pic2	99.6192	33.5636
Pic3	99.6022	33.6622
Pic4	99.6233	33.4634
Pic5	99.6334	33.5642

NPCR has the following range from 0 to 1. When NPCR (T^1, T^2) is equal to zero it indicates that every pixel has identical values for T^1 and T^2 while UACI (T^1, T^2) is equal one, it indicates that every pixel of T^1 are updated as compared to the values of T^2. The critical numbers are used in both the NPCR and UACI tests. The gray value of the

encrypted picture is $T^1(i,j)$, whereas the grey value in the latest encrypted picture is $T^2(i,j)$ [28]. In Table 3, a comparison of two metrics which are NPCR and UACI for the proposed technique.

5.5 Keyspace

The large keyspace prevents exhaustive searches for keys. The challenge of identifying the precise key value can be resolved by testing a variety of values until the appropriate value is determined.

Four independent variables are used in the proposed technique: two positive coefficients, l and k, and two secret initial values (x_0 and x_1). The number of different values exceeds (10^{14}) because these variables are double-precision integers.

A keyspace of $(10^{14})^4 = 10^{56} \approx 2^{186}$ besides, the proposed algorithm adopts an *IV* of 128 bits as the key which provides a total key space of ($2^{186} \times 2^{128} = 2^{314}$) for a single block.

Therefore, a comparison of the key space for the suggested technique and other relevant techniques is shown in Table 4. The outcomes show that the technique can successfully defend against brute force attacks by generating the input sequence from a nonlinear chaotic structure while maintaining the flexibility of this algorithm.

The main characteristics of this technique are the intrinsic nonlinearity, unpredictability, and quasirandom properties of chaotic systems to prevent linear attacks.

Table 4. Comparative analysis of the keyspace

Encryption Method	Keyspace
Li et al. [2]	2^{100}
Öztürk and Kılıç [10]	2^{128}
Shakiba [11]	2^{128}
Louzzani et al. [14]	$2^{292.93}$
Stoyanov [29]	2^{157}
Proposed Algorithm	2^{314} for each generated random sequence of 128 bits

5.6 Tests of the NIST SP 800-22

The degree of randomness of the encryption was tested using NIST SP 800-22 to analyze the chaotic nature of the random sequence produced by the proposed random generator. It relies on two initial keys (x_0, x_1) and two parameters (k, l) and generates as many random bits as necessary by the IV. Fifteen tests were created for this study to determine whether a sequence was random.

Table 5 contains a set of test results and in Table 6, the proposed approach is compared with previous research using the "Lena" picture.

For an individual test, the p-value > 0.01 means that the sequence would be considered to be random. As noted in Table 5, all p-values for the four sequences are larger than 0.01 this means they pass the NIST test successfully.

Table 5. Results of NIST SP 800-22 tests

Test	Sequnce 1	Sequnce 2	Sequnce 3	Sequnce 4	Pass/Fail
Frequency	0.4156175	0.3783561	0.6351866	0.7325413	All Pass
Frequency analysis inside a block	0.7034011	0.7536041	0.5028178	0.5372379	All Pass
Test for the extended run of ones in each block	0.2149633	0.6383713	0.6126157	0.0303386	All Pass
Runs	0.5016738	0.161039	0.8875099	0.6349206	All Pass
Discrete Fourier transform	0.4917503	0.3249381	0.9601341	0.7656526	All Pass
Overlapping template matching test	0.7459741	0.8192108	0.8729810	0.4972001	All Pass
Matching non-overlapping templates	0.9646992	1.0000128	0.9987128	0.9871238	All Pass
The rank of a binary matrix	0.6226403	0.7777956	0.8301560	0.1363148	All Pass
Overlapping templates that match	0.2047836	0.4738326	0.2325743	0.3623905	All Pass
"Universal Statistical" by Maurer	0.7248827	0.1214450	0.4474011	0.8798172	All Pass
Serial test	0.5273036	0.1377411	0.0112511	0.1363323	All Pass
Complexity of linear processes	0.1066874	0.3192507	0.3943641	0.2879147	All Pass
Approximate entropy	0.9306489	0.1851236	0.0126257	0.4873995	All Pass
Variation of random excursions	0.1363367	0.0082518	0.1342495	0.2142088	All Pass
Random excursions	0.4644700	0.0429001	0.0376354	0.2353400	All Pass
Total sums	0.3308419	0.437878	0.4665543	0.8146067	All Pass

Table 6. Comparison of the proposed approach with previous research utilizing the "Lena" picture

References	UACI	NPCR	Entropy	C.C		
				Diagonal	Vertical	Horizontal
[11]	50.0116	99.6115	7.9991	0.01399	0.01391	0.01534
[16]	33.46	99.60	7.9998	0.00001	-0.0004	0.0005
[18]	33.41	99.61	N/A	N/A	−0.0035	N/A
[23]	33.57	99.62	7.9992	-0.0006	0.0028	-0.0016
[25]	33.45	99.56	7.9993	-0.0027	-0.0020	0.0048
[28]	33.42	99.59	7.9891	-0.0022	0.0171	-0.0028
[30]	30.35	99.62	7.999	-0.00099	-0.00018	0.0018
[31]	33.45	99.61	7.9993	0.0077	-0.0293	-0.0031
[32]	N/A	N/A	7.9968	-0.0049	-0.0104	0.0029
Suggested Method	33.46	99.64	**7.9993**	**-0.00055**	**-0.00065**	**-0.00044**

Although the standard image `Lena' is non-medical image but a comparison is done in Table 6 with the related works that implement their methods on standard image `Lena'. This comparison is necessary to justify the robustness of the proposed system. We compared with standard Lena image because it is difficult to find related works implement their

suggested method on the same medical images that are used by our proposed system.

Table 7 shows performance speeds of AES, RC6 and proposed method on gray scale images of different sizes of the MRI image with secret key length 128-bits. As we nots that the proposed algorithm is more complex due to the use of more rounds and operations. This adds overhead and makes proposed method slower, especially on software implementations.

Table 7. Performance speed of the proposed method with AES and RC6

Size in Byte	Tine in Sec. AES	Tine in Sec. RC6	Time in Proposed Method
17.1	48.3384	13.6542	29.3048
65 kb	565.7291	142.2473	293.4946
256 kb	7872.1891	1978.2874	4054.5694

6. CONCLUSION

This research emphasized the necessity of a secure cryptographic approach to maintain confidential data in medical images and establish the appropriate level of security within healthcare facilities. A new picture encryption technique is developed by two-dimensional Chebyshev polynomial mappings and RC6 algorithm.

The suggested approach depends on a 128-bit pseudo-bit random generator using two-dimensional Chebyshev polynomial mappings and RC6. A randomness analysis is used to acquire NIST measurements, histograms, Correlation Coefficients, and entropy. UACI and NPCR are utilized to calculate the reliability and variation of image encryption. This approach performs better than previous image encryption solutions due to increased security with an efficient generating performance and an acceptable arbitrary pattern.

REFERENCES

[1] Masood, F., Driss, M., Boulila, W., Ahmad, J., Rehman, S.U., et al. (2022). A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations. Wireless Personal Communications, 127(2): 1405-1432. https://doi.org/10.1007/s11277-021-08584-z

[2] Li, M., Pan, S., Meng, W., Guoyong, W., Ji, Z., Wang, L. (2022). Medical image encryption algorithm based on hyper-chaotic system and DNA coding. Cognitive Computation and Systems, 4(4): 378-390. https://doi.org/10.1049/ccs2.12070

[3] Gafsi, M., Abbassi, N., Hajjaji, M.A., Malek, J., Mtibaa, A. (2020). Improved chaos-based cryptosystem for medical image encryption and decryption. Scientific Programming, 2020(1): 6612390. https://doi.org/10.1155/2020/6612390

[4] Imdad, M., Ramli, S.N., Mahdin, H. (2022). An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys. Symmetry, 14(3): 604. https://doi.org/10.3390/sym14030604

[5] Al-Saadi, H.M., Alshawi, I. (2023). Provably-secure led block cipher diffusion and confusion based on chaotic maps. Informatica, 47(6): 105-114. https://doi.org/10.31449/inf.v47i6.4596

[6] Naik, R.B., Singh, U. (2024). A review on applications of chaotic maps in pseudo-random number generators and encryption. Annals of Data Science, 11(1): 25-50. https://doi.org/10.1007/s40745-021-00364-7

[7] Zhang, P., Hu, H., Hu, X., Yang, X. (2017). New pseudorandom number generators from block ciphers. In 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), Shenzhen, China, pp. 162-171. https://doi.org/10.1109/DSC.2017.22

[8] Ragab, A.A.M., Madani, A., Wahdan, A.M., Selim, G.M. (2023). Design, analysis, and implementation of a new lightweight block cipher for protecting IoT smart devices. Journal of Ambient Intelligence and Humanized Computing, 14(5): 6077-6094. https://doi.org/10.1007/s12652-020-02782-6

[9] Saraswathi, P.V., Venkatesulu, M. (2017). A novel stream cipher using pesudo random binary sequence generator for medical image encryption. In 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, pp. 425-429. https://doi.org/10.1109/ICOEI.2017.8300961

[10] Öztürk, İ., Kılıç, R. (2018). A new pseudo random number generator based on Chebyshev maps and parameter switching. In 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, pp. 1-3. https://doi.org/10.1109/CEIT.2018.8751842

[11] Shakiba, A. (2019). A novel randomized one-dimensional chaotic Chebyshev mapping for chosen plaintext attack secure image encryption with a novel chaotic breadth first traversal. Multimedia Tools and Applications, 78(24): 34773-34799. https://doi.org/10.1007/s11042-019-08071-5

[12] Tutueva, A.V., Nepomuceno, E.G., Karimov, A.I., Andreev, V.S., Butusov, D.N. (2020). Adaptive chaotic maps and their application to pseudo-random numbers generation. Chaos, Solitons & Fractals, 133: 109615. https://doi.org/10.1016/j.chaos.2020.109615

[13] Krishnamoorthi, S., Jayapaul, P., Dhanaraj, R.K., Rajasekar, V., Balusamy, B., Islam, S.H. (2021). Design of pseudo-random number generator from turbulence padded chaotic map. Nonlinear Dynamics, 104(2): 1627-1643. https://doi.org/10.1007/s11071-021-06346-x

[14] Louzzani, N., Boukabou, A., Bahi, H., Boussayoud, A. (2021). A novel chaos based generating function of the Chebyshev polynomials and its applications in image encryption. Chaos, Solitons & Fractals, 151: 111315. https://doi.org/10.1016/j.chaos.2021.111315

[15] Sreedharan, S., Eswaran, C. (2021). A lightweight encryption scheme using Chebyshev polynomial maps. Optik, 240: 166786. https://doi.org/10.1016/j.ijleo.2021.166786

[16] Dridi, F., El Assad, S., El Hadj Youssef, W., Machhout, M., Lozi, R. (2022). Design, implementation, and analysis of a block cipher based on a secure chaotic generator. Applied Sciences, 12(19): 9952. https://doi.org/10.3390/app12199952

[17] Rashidi, B. (2024). Lightweight structure of random key generation for PRESENT block cipher. International Journal of Industrial Electronics Control and Optimization, 7(1): 41-51. https://doi.org/10.22111/ieco.2024.47375.1511

[18] Zhu, S., Deng, X., Zhang, W., Zhu, C. (2023). Image

encryption scheme based on newly designed chaotic map and parallel DNA coding. Mathematics, 11(1): 231. https://doi.org/10.3390/math11010231

[19] Kocarev, L., Makraduli, J., Amato, P. (2005). Public-key encryption based on Chebyshev polynomials. Circuits, Systems and Signal Processing, 24(5): 497-517. https://doi.org/10.1007/s00034-005-2403-x

[20] Ji, X., Chen, Y., Yang, W., Wu, Q. (2023). Security and data encryption effect of high ciphertext based on improved RC6 algorithm for WSN. Results in Physics, 53: 106959. https://doi.org/10.1016/j.rinp.2023.106959

[21] Helmy, M., El-Rabaie, E.S.M., Eldokany, I.M., El-Samie, F.E.A. (2017). 3-D image encryption based on Rubik's cube and RC6 algorithm. 3D Research, 8(4): 38. https://doi.org/10.1007/s13319-017-0145-8

[22] Shimal, A.F., Helal, B.H., Hashim, A.T. (2021). Extended of TEA: A 256 bits block cipher algorithm for image encryption. International Journal of Electrical and Computer Engineering, 11(5): 3996. https://doi.org/10.11591/ijece.v11i5.pp3996-4007

[23] Salman, L.A., Hashim, A.T., Hasan, A.M. (2022). Selective medical image encryption using polynomial-based secret image sharing and chaotic map. International Journal of Safety and Security Engineering, 12(3): 357-369. https://doi.org/10.18280/ijsse.120310

[24] Jirjees, S.W., Yousif, N.A., Hashim, A.T. (2022). Colour image privacy based on cascaded design of symmetric block cipher. Journal of Engineering Science and Technology, 17(3): 2135-2156.

[25] Wang, X., Zhao, H., Feng, L., Ye, X., Zhang, H. (2019). High-sensitivity image encryption algorithm with random diffusion based on dynamic-coupled map lattices. Optics and Lasers in Engineering, 122: 225-238. https://doi.org/10.1016/j.optlaseng.2019.04.005

[26] Mao, Y., Chen, G., Lian, S. (2004). A novel fast image encryption scheme based on 3D chaotic baker maps. International Journal of Bifurcation and Chaos, 14(10): 3613-3624. https://doi.org/10.1142/S021812740401151X

[27] Chen, G., Mao, Y., Chui, C.K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals, 21(3): 749-761. https://doi.org/10.1016/j.chaos.2003.12.022

[28] Chen, X., Hu, C.J. (2017). Adaptive medical image encryption algorithm based on multiple chaotic mapping. Saudi Journal of Biological Sciences, 24(8): 1821-1827. https://doi.org/10.1016/j.sjbs.2017.11.023

[29] Stoyanov, B. (2014). Pseudo-random bit generation algorithm based on Chebyshev polynomial and Tinkerbell map. Applied Mathematical Sciences, 8(125): 6205-6210. https://doi.org/10.12988/ams.2014.48676

[30] Alexan, W., Chen, Y.L., Por, L.Y., Gabr, M. (2023). Hyperchaotic maps and the single neuron model: A novel framework for chaos-based image encryption. Symmetry, 15(5): 1081. https://doi.org/10.3390/sym15051081

[31] Ayubi, P., Setayeshi, S., Rahmani, A.M. (2020). Deterministic chaos game: A new fractal based pseudo-random number generator and its cryptographic application. Journal of Information Security and Applications, 52: 102472. https://doi.org/10.1016/j.jisa.2020.102472

[32] Moysis, L., Tutueva, A., Volos, C.K., Butusov, D. (2020). A chaos based pseudo-random bit generator using multiple digits comparison. Chaos Theory and Applications, 2(2): 58-68.