

A Hyper-chaotic Medical Image Encryption with Optimized Key Value

Subhajit Das*

University of Kalyani, West Bengal, India

E-mail: subhajit.batom@gmail.com

ORCID iD: <https://orcid.org/0000-0003-3759-0975>

*Corresponding author

Manas Kumar Sanyal

University of Kalyani, West Bengal, India

E-mail: manas_sanyal@rediffmail.com

ORCID iD: <https://orcid.org/0000-0002-5760-8207>

Received: 10 December 2023; Revised: 14 January 2024; Accepted: 15 March 2024; Published: 08 June 2024

Abstract: This article delves into a medical image encryption/decryption method based on hyper chaotic dynamics and genetic algorithms. The proposed algorithm boasts simplicity in implementation, featuring straightforward operations that render it well-suited for real-time applications while elevating its security measures. Leveraging the sensitivity of chaotic behaviour to initial conditions, a genetic algorithm is employed to select optimal initial conditions for the 5D multi-wing hyper-chaotic system. Initially, a secret key generation method based on the input image is applied, followed by stages of diffusion and encryption utilizing the chaotic system. The secret key undergoes optimization through a genetic algorithm, considering specific parameters within the encrypted image as encryption factors. Subsequently, the encrypted image with the optimized secret key is finalized, serving as the basis for decrypting the cipher image. The proposed method undergoes simulation, testing, and comparison against other image encryption algorithms. Both experimental results and computer simulations affirm the robustness of this cryptographic system, showcasing a significant key space value (2^{256}), high key sensitivity (Number of Pixels Change Rate: NPCR > 99.55%, Unified Average Changing Intensity: UACI > 33.37%), and its ability to fend off various types of attack.

Index Terms: Secret Key Value, Multipoint Crossover, Autonomous Hyper-chaotic, Optimization, Decryption Key.

1. Introduction

Telehealth care services play an essential role in the current pandemic situation. These healthcare services include telemedicine, tele surgery, tele radiology, etc. The basis of telemedicine is medical image transformation from patient end to doctor. This procedure helps the doctor with the proper diagnosis. Medical images like Magnetic resonance imaging (MRI), computed tomography (CT), ultrasound, X-ray, etc, provide a visual representation of body organs and tissue that aids in diagnosis and proper treatment planning [1]. In this scenario, medical images are incredibly crucial for correct treatment. The Internet and combining different medical images radically change the healthcare system [2]. In modern healthcare systems, diagnostic images with patients' personal confidential information are stored and transmitted through public unsecured networks [3]. Hence, a lot of unique isolation is needed to be controlled either in transit or in storage.

A more direct and proper approach that protects medical images against illegal information leakage is known as image encryption [4,5]. This process converts an image from its original form to an unrecognizable form using a secret key. However, the medical image encryption algorithm differs from general image encryption because medical images have large amounts of data with solid pixel correlation. Moreover, medical images are very much more sensitive than regular images. As security is the main issue, different mathematical and scientific approaches are applied to image encryption.

Chaos-based image encryption algorithms are widely popular among researchers. Chaotic systems are susceptible to initial conditions. A tiny change in the initial condition generates an entirely different output. Chaos sequences are applied in different encryption algorithm phases to enhance the results. Permutation and diffusion stages are efficiently managed by generating extremely random numbers through chaos.

Dai et al. [6] demonstrated a medical image encryption based on logistic maps and Chebyshev maps. Logistic map-based encryption could be more efficient due to low security and key space. They have overcome this problem by using

their proposed maps, but it is observed that their algorithm needs to be more secure if logistic values are correctly set.

C. Fu et al. [7]. and J. Chen et al. [8] introduced a secure medical image encryption based on chaos theory. They used a level shuffling algorithm in the permutation phase. It is observed that the appropriate security level was achieved after executing fewer encryption rounds. Still, a one-dimensional chaotic system must be fixed with small secret key space problems and weak security. Wang et al. [9] combined bit-plane decomposition and chaotic systems in their proposed medical image encryption scheme. The experimental outcomes of their proposed algorithm had satisfactory results in all standard tests. Chen and Hu [10] introduced a hyper-chaotic map to ensure the least security. The Logistic sine chaos map was employed in their suggested approach to scramble the plain image. After that, the scrambled image is separated into predetermined subblocks, which are conveniently encrypted using a hyper-chaotic technique. However, it is necessary to solve two chaotic system logistic maps and hyper-chaotic maps that increase the complexity of the proposed algorithm.

Li et al. [11] developed an orbit perturbation and dynamic state variable selection system to demonstrate the effectiveness of chaos-based image encryption. The algorithm is based on a two-dimensional logistic-adjusted sine map, which generates highly random values. A previously handled pixel continuously interrupts the mapping orbit to select one or two state variables to form the key stream. The proposed algorithm's performance was evaluated and discovered to provide adequate functionality. Shanshan Li et al. [12] proposed a two-dimensional chaotic map in their encryption scheme. The research paper represents a novel medical image encryption based on 2D zigzag confusion and dynamic diffusion. This approach offers a more straightforward structure and reduced time complexity than high-dimensional chaotic systems. However, compared to low-dimensional chaotic maps, their structure becomes more intricate, making it more challenging to predict and analyze. This algorithm is applicable for encrypting grayscale images of any size.

Javan et al. [13] demonstrate a multi-mode synchronization of hyperchaotic system-based novel medical image encryption. This method addresses the challenge of achieving synchronization in hyper-chaotic systems for secure image encryption. An adaptive robust controller is designed to synchronize chaotic systems with variable parameters in the presence of disturbance and uncertainty. Shuang Liu et al. [14] propose a novel medical image protection combining stream cipher and chaos theory. This proposed method employs the Chebyshev map to generate the encryption key using stream cipher principles. Through a series of coding operations and initial value settings, the image undergoes chaotic processing. Encryption involves logical mapping in X and Y dimensions, ensuring comprehensive coverage.

Considering the research mentioned above papers, chaos theory plays a crucial part in image encryption. But it [15-17] has been analysed that chaos-based encryption has some security weaknesses. In most cases, it was found that chaos-based algorithms cannot efficiently resist chosen plain text and cipher text attacks. In this scenario, we are interested in a hyper-chaotic system to eliminate the weakness of a low-order chaotic system. Furthermore, it is also observed that the chaotic system's performance is based on initial conditions so that unfitting initial conditions may lead to weak performance of the encryption algorithm. Optimizing proper initial conditions is a challenging issue for researchers.

Many researchers take hash values for generating secret keys, but it is very time-consuming when the input image size is large enough.

Inspired by the situation, we have developed a medical image cryptosystem based on a 5D multi-wing hyper-chaotic system. We produce a practical key value that entirely depends on the input image to resist different kinds of attacks. The contributions of this paper are as follows

- A highly secret key is generated from the input image to ensure the sensibility of the proposed algorithm
- Initial parameters of the chaotic system are optimized by applying the genetic algorithm.
- We start encryption with a secret key value, but the optimized secret key is responsible for image retrieval in the decryption stage.
- To bypass the harmful effect of the transitional procedure, we iterate a chaotic system with a number that is not fixed like another existing algorithm. Instead, it depends upon the secret key value.
- The permutation is done at the bit and value levels for better results.

Our proposed method's encryption key and decryption fundamental values are slightly different. The performance of our proposed algorithm is proved by carrying out various statistical and mathematical tests. The orientation of the rest of the paper is as follows: Section 2 introduces GA and the hyper-chaotic system. The proposed method is then given in Section 3, and a step-by-step algorithm is described in Section 4. Section 5 then simulates and assesses the outcomes. Finally, Section 5 concludes the paper with overall findings and future work.

2. Preliminaries

2.1. Genetic Algorithm

Genetic algorithms (GA) are computational models for search and optimization-based problems. They are generally applied to NP-Hard problems. The basic principles of GA were first developed by John Holland [18]. These algorithms are inspired by the principle of natural selection and survival for the fittest, which Darwin introduced in "The Origin of Species". Initially, a population (a subset of solutions) is taken randomly or by other heuristics. These solutions are known as chromosomes. Then, a fitness function is calculated, which takes possible solutions of a given problem as input and

produces a score for each solution that defines how good the solution is. Parents are selected depending on their fitness. Then, parents with high fitness are allowed to mate, and crossover and mutations are performed to produce new offspring, which contain the characteristics of both fit parents. The less-fit solutions are less likely to get selected and are not allowed for mating. So, a better population is produced in the next generation by mating the fit parents. Thus, in every generation, a better population is achieved. A well-designed GA is likely to converge to an optimal solution.

2.2. Diagonal Multipoint Crossover

In our proposed work, we use diagonal crossover where n crossover points and n+1 offspring are produced from n+1 parents [19]. According to our proposed method, we have used four crossover points with five parents, which makes five children. We have used four crossover points after the 4th bit, 6th bit, 8th bit, and 10th bit, respectively, on the 12-bit parent chromosome. The first 4 bits of the parent are copied to the respective child. The values of the rest of the segments of a child are copied, successively taking the values of the following parents to the next segment. So, each child inherits genes from all parents. In Fig 1, the diagonal crossover, which we have used in our proposed method, is explained

parent 1	5	7	8	A	C	B	7	B	C	D	1	2
parent 2	A	B	3	4	5	6	8	9	8	A	C	F
parent 3	C	A	9	8	A	B	C	C	0	7	1	3
parent 4	6	7	9	C	F	5	6	9	1	9	8	1
parent 5	2	3	0	5	1	9	2	1	3	F	A	C
child 1	5	7	8	A	5	6	C	C	1	9	A	C
child 2	A	B	3	4	A	B	6	9	3	F	1	2
child 3	C	A	9	8	F	5	2	1	C	D	C	F
child 4	6	7	9	C	1	9	7	B	8	A	1	3
child 5	2	3	0	5	C	B	8	9	0	7	8	1

Fig.1. Diagonal crossover for 5 parents

2.3. 5-D Autonomous Hyper-chaotic

Amin Zarei[20] has proposed a 5-D autonomous hyper-chaotic attractor with complex dynamics where eight parameters are there with one equilibrium point. Four-wing hyper-chaotic and chaotic attractors can be produced by new dynamics under certain initial conditions and parameter settings. The proposed system's nonlinear characteristics differ from most well-known chaotic and hyper-chaotic attractors. Authors [20] applied symmetry properties about the coordinates, initial conditions, and bifurcation diagrams to design multi-scroll or multi-wing systems in the proposed chaotic system. The multiple attractor phenomenon has been presented, and as per the phenomenon, many strange attractors are observed. When subjected to parameter changes, the chaotic system's outcome to variations in initial conditions and its dynamic characteristics has been thoroughly explored. Key aspects analysed include the Lyapunov exponent's spectrum, bifurcation diagrams, and phase portraits. These simulations offer valuable insights into the system's unique chaotic nature and sensitivity to initial states, shedding light on its intricate long-term behaviour and complex attractor structures. Differential equations for a 5-dimensional system are given below [20]:

$$\dot{x}_1 = -ax_1 + x_2x_3 \tag{1}$$

$$\dot{x}_2 = -bx_2 + fx_5 \tag{2}$$

$$\dot{x}_3 = -cx_3 + gx_4 + x_1x_2 \tag{3}$$

$$\dot{x}_4 = dx_4 - hx_3 \tag{4}$$

$$\dot{x}_5 = ex_5 - x_2x_1^2 \tag{5}$$

Where, state variables are x_1, x_2, x_3, x_4, x_5 and the real constant parameters of the system are a, b, c, d, e, f, g, h . x_2x_3, x_1x_2 and $x_2x_1^2$ are the non-linear terms in the dynamical system. Fourth order Runge-Kutta method is used to solve the system when $a = 10, b = 60, c = 20, d = 15, e = 40, f = 1, g = 50, h = 10$. The equations are solved with provided initial conditions and different constant by Runge_Kutta method. Different three dimension and two-dimension diagrams as defined in fig 2 demonstrate that the said chaotic system is very effective in image encryption method.

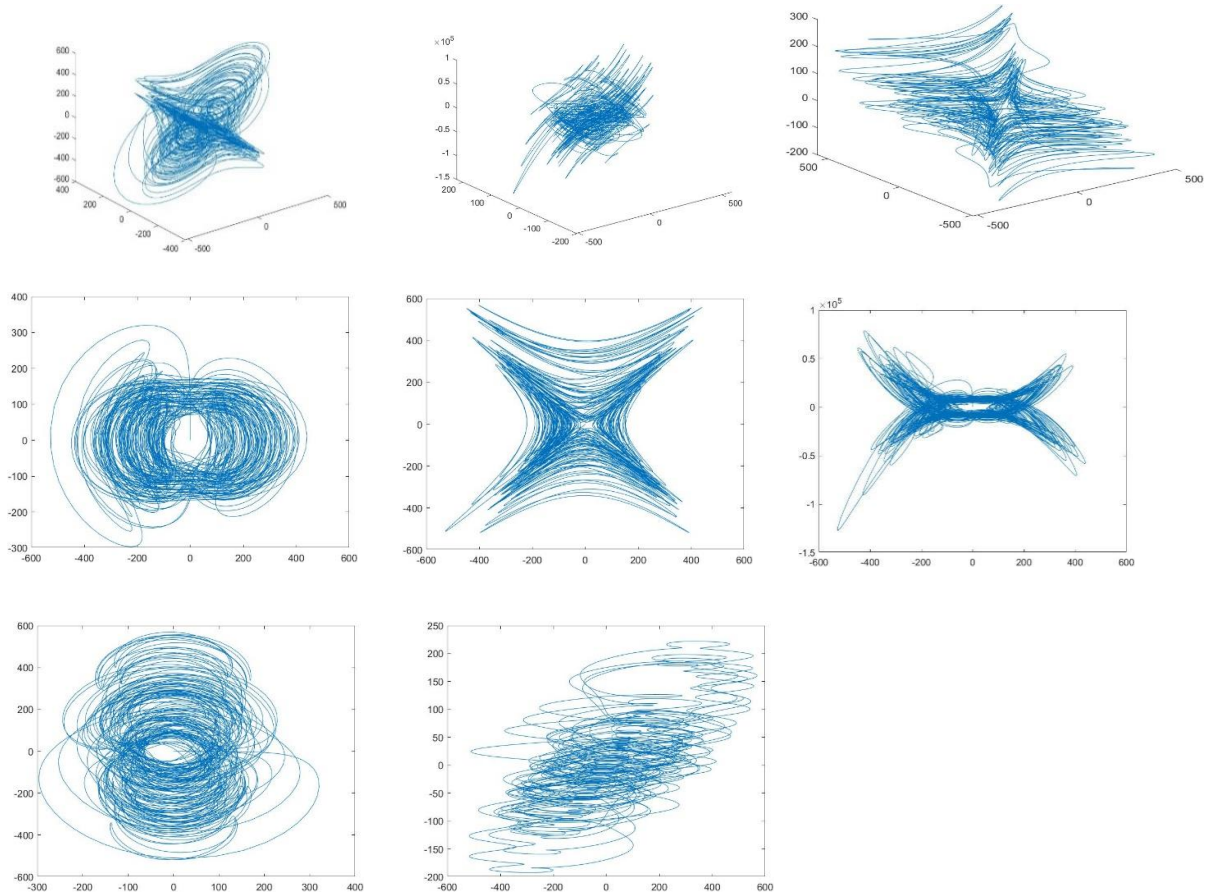


Fig.2. Different 2D and 3D representation of chaotic characteristics with $x(1)=x(2)=0, x(3)=x(4)=x(5)=1$, $a = 10, b = 60, c = 20, d = 15, e = 40, f = 1, g = 50, h = 10$, with (x_1, x_2, x_3) plane, (x_3, x_4, x_5) plane, (x_1, x_3, x_4) plane, (x_1, x_2) plane, (x_1, x_3) plane, (x_1, x_5) plane, (x_2, x_3) plane and (x_3, x_4) plane

3. Methods

Our medical image cryptosystem consists of different parts: key generation, initial value generation, pixel shuffling, encryption, and, finally, optimization of the initial value.

Many researchers use the hash value of the input image to resist the chosen plain text attacks. Still, this process is only appropriate when the measurement of the image is manageable, and it takes a long waiting time to change all entered pixel values to their corresponding hash value. Again, sizeable key space is vital for an efficient encryption algorithm. To overcome the problem, this paper introduces a unique key generation process that generates an exceptional 256-bit value depending upon plaintext image.

The 256-bit generated values are passed through a straightforward algorithm to generate the initial values of control parameters. Hence, the Runge Kutta method of order four is used to solve five differential equations using the newly developed control parameter. The diffusion process-based model faces a significant problem [21,22], especially in medical images, so to overcome the problem, our cryptosystem uses pixel shuffling and encryption phases. To change the orientation of the pixel position, first of all, we circularly rotate every row and column with a specified number received from random sequences, then shuffle their positions according to the random number sorting procedure. To change the pixel value, we circularly rotate each pixel value with a specified number and finally operate a bitwise xor operation that depends on the present input and the previous stage's output.

Encryption Factor (E_t)

As the response of the chaotic system extraordinarily relies upon initial values, we tune the preliminary values of the system using a genetic algorithm. Initial values are selected as the initial population. Using these values, the proposed algorithm generates an encrypted image. Different static values of this encrypted image are stored. In the subsequent iteration, genetic algorithms generate new initial values and follow a similar process to achieve a new encrypted image with new statistical values. The complete process iterates up to a specified number of iterations. Lastly, the initial value for which the best encryption result is obtained is considered the final key, and its associated output image is selected as the encrypted image. A stepwise flow diagram of the proposed system is presented in Fig 3.

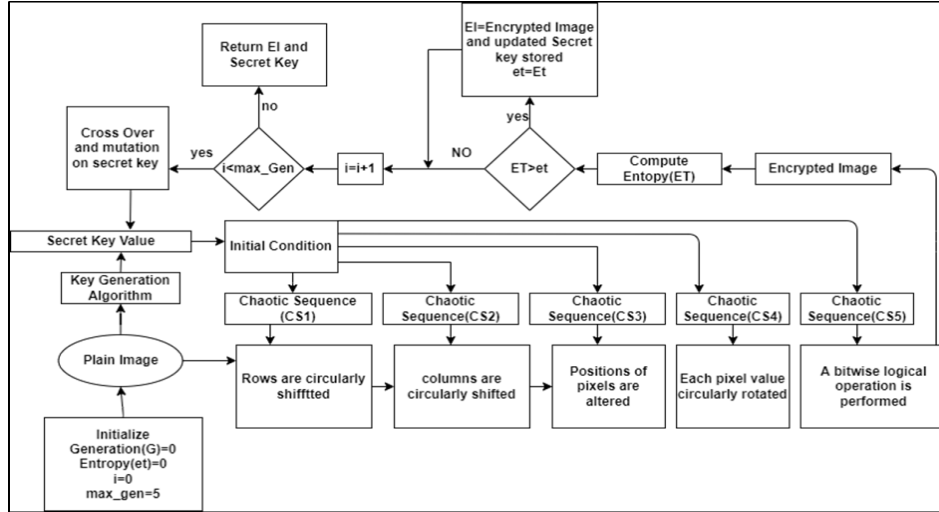


Fig.3. Flow diagram of proposed method

4. Optimized Medical Image Encryption Algorithm

4.1. Key Generation for Encryption

This phase generates input image based 256 bit number

Step 1: Create a random binary sequence of 256 bits, then translate it into a 64-hexadecimal equivalent.

Step 2: Divide 64 hexadecimal number into four equals groups where each group contains 16 hexadecimal number.

$$K = \{k_1, k_2, k_3, k_4\} \text{ where each } k_i \text{ is consist of 16 hexadecimal number} \quad (6)$$

Step 3: Divide input image $img_{m \times n}$ into 4 parts and computes the average value of each part.

$$s_1 = \text{floor}(\text{avg}(img_{i,j})) \text{ where } 1 \leq i \leq \frac{m}{2} \text{ and } 1 \leq j \leq \frac{n}{2}; \quad (7)$$

$$s_2 = \text{floor}(\text{avg}(img_{i,j})) \text{ where } 1 \leq i \leq \frac{m}{2} \text{ and } (\frac{n}{2} + 1) \leq j \leq n; \quad (8)$$

$$s_3 = \text{floor}(\text{avg}(img_{i,j})) \text{ where } \frac{m}{2} + 1 \leq i \leq m \text{ and } 1 \leq j \leq \frac{n}{2}; \quad (9)$$

$$s_4 = \text{floor}(\text{avg}(img_{i,j})) \text{ where } \frac{m}{2} + 1 \leq i \leq m \text{ and } (\frac{n}{2} + 1) \leq j \leq n; \quad (10)$$

Step 4: Covert these average values into its equivalent hexadecimal number and take first 16 hexadecimal number. If the length of hexadecimal value is less than 16 then zeros are added at the end to make a string of 16 hexadecimal number.

$$s'_i = \text{dec2hex}(\text{floor}(s_i)) \text{ where } 1 \leq i \leq 4 \quad (11)$$

Step 5: Perform a bitwise xor operation between each k_i and s'_i and concatenate the 4 groups results to obtain effective key(eK).

$$key_i = \text{bitwisexor}(k_i, s'_i) \text{ where } 1 \leq i \leq 4 \quad (12)$$

$$eK = \{key_1, key_2, key_3, key_4\} = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{64}\} \quad (13)$$

4.2. Initial Values of the Chaotic Equations and Solving the Equations

The initial values of chaotic equations have a significant impact on every aspect of the crypto system. Moreover, producing appropriate initial values that satisfying the condition of chaotic system is really a challenging task. As per 5D autonomous hyper-chaotic system the initial values can be obtained as follows

$$\begin{aligned} x_1 &= \text{hex_to_decimal}(\text{substring}(1,12, eK) \times 10^{-16}) \\ x_2 &= \text{hex_to_decimal}(\text{substring}(13,24, eK) \times 10^{-16}) \\ x_3 &= \text{hex_to_decimal}(\text{substring}(25,36, eK) \times 10^{-16}) \end{aligned} \quad (14)$$

$$\begin{aligned}x_4 &= \text{hex_to_decimal}(\text{substring}(37,48, eK) \times 10^{-16}) \\x_5 &= \text{hex_to_decimal}(\text{substring}(49,60, eK) \times 10^{-16}) \\x_6 &= (10^3 - \text{rem}(\text{hex_to_decimal}(\text{substring}(61,64, eK)), 10^2))\end{aligned}$$

Here $\text{substring}(x,y,\text{length})$ function generates a string from x location to y from string 'length'. $\text{hex_to_decimal}(x)$ function generates decimal equivalent of hexadecimal number x and $\text{rem}(x,y)$ generates remainder part when x is divide by y .

A chaotic system generates an extraordinary random sequence so a number of iterations play an important role. Most of the researchers iterate chaotic systems by 1000 times or by a fixed number to remove their adverse effect. To remove that our system iterate chaotic system with a specified number that generates from the effective key. Here x_1, x_2, x_3, x_4, x_5 is used as initial value of chaotic equation and x_6 is used to iterate each chaotic equation. Utilizing the initial values we solve the system equations (1-6) by Runge Kutta method (order 4) and generate five different vectors v_1, v_2, v_3, v_4 , and v_5 of length $m \times n$ each. Moreover, to eliminate the harmful impact of the transitional procedure we iterate the system x_6 times.

4.3. Pixel Shuffling and Encryption

Step 1: each row is circularly rotated according to the value taking from v_1 sequentially.

$$v1' = \text{rem}(|v1 \times 10^{12}|, n); \quad (15)$$

$$\text{img1} = \text{circularshift}(\text{img}(i,:), v1'_k), \forall i = 1, 2, \dots, m; k = 1, 2, \dots, m \quad (16)$$

Step 2: each column is circularly rotated according to the value taking from v_2 sequentially.

$$v2' = \text{rem}(|v2 \times 10^{12}|, m); \quad (17)$$

$$\text{img2} = \text{circularshift}(\text{img1}(:,j), v2'_k), \forall j = 1, 2, \dots, n; k = 1, 2, \dots, n \quad (18)$$

Step 3: Next convert img2 into one dimensional array ($\text{img2}'$). $\text{img2}'$ is sorted and stored in $v3'$. Now value positions of $v3'$ in $v3$ is searched and stored in loc . The pixels of one-dimensional array are changed according to loc and after that it is reshaped into two dimensions.

$$\text{img2}' = \text{oneD}(\text{img2}) \quad (19)$$

$$v3' = \text{sort}(\text{img2}') \quad (20)$$

$$\text{img3}'(k) = \text{img2}'(\text{loc}(k)), \quad k = 1, 2, 3, \dots, m \times n \quad (21)$$

$$\text{img3} = \text{twoD}(\text{img3}', m, n) \quad (22)$$

Here $\text{oneD}()$ function converts a two-dimension array in to one dimension. $\text{twoD}(x, m, n)$ convert x into two-dimension array of m rows and n columns

Step 4: Each pixel of img3 is circularly rotated according to the values taken from v_4 .

$$v4' = \text{rem}(|v4 \times 10^{12}|, 8); \quad (23)$$

$$\text{img4} = \text{circularshift}(\text{img}_{i,j}, v4'_k), \forall i = 1, 2, \dots, m; j = 1, 2, \dots, n; k = 1, 2, \dots, m \times n \quad (24)$$

Step 5: A bit wise XOR is performed using v_5 and encrypted image EI is obtained.

$$\text{img4}' = \text{oneD}(\text{img4}) \quad (25)$$

$$v5' = \text{rem}(|v5 \times 10^{12}|, 256) \quad (26)$$

$$EI(i) = \text{img4}'(i) \oplus EI(i-1) \oplus v5'(i), i = 1, 2, 3, \dots, m \times n \text{ and } EI(0) = 1 \quad (27)$$

$$\begin{aligned}\text{Encrypted}_{\text{image}} &= \text{twoD}(EI, m, n) \\ \text{Encrypted}_{\text{image}} &= \text{construct_twoD}(EI)\end{aligned} \quad (28)$$

4.4. Optimization

In our proposed algorithm most popular optimization technique Genetic algorithm is used for choosing appropriate initial values. Here five initial values (60 hexadecimal numbers) of hyper-chaotic system i.e x_1, x_2, x_3, x_4, x_5 are used as an initial population. Using these values image shuffling and encryption is done as stated above. Then entropy, three types of correlation coefficient, NPCR and UACI of the encrypted image is computed and stored.

In subsequent levels, diagonal multi-point crossover is applied on $x_i (\forall 0 \leq i \leq 4)$. For this 4th bit, 6th bit, 8th bit and 10th position are selected as crossover point as described in sec 2.1. In mutation phase change the MSB of 5th and 9th bit of newly generated offspring. After crossover and mutation new initial values are obtained. Using these new initial values encrypted image is created and its statistical values like entropy, three types of correlation coefficient, NPCR and UACI values are computed in a similar fashion.

All these said characteristics are combined into a single value by using factor analysis. In this process we determine a factor that multiply with each characteristic to convert it into a single value. That single value is noted as encryption factor (Ef) in our proposed algorithm. To determine the encryption factor proposed algorithm is applied to thirty sets of samples. As per ref [23] it is noted that the ideal entropy value is closer to 8, NPCR value is 99.6094 and UACI value is 33.6435. Hence to enhance the effectiveness of our proposed algorithm our main focus is to

$$\min\{abs(8 - entropy), abs(66.6094 - NPCR_{value}), abs(33.6435 - UACI_{value})\} \tag{29}$$

Besides this we also minimize the horizontal, vertical and diagonal correlation coefficient. The sample results that use for factor analysis is furnished in Table 1.

Table 1. Values of different characteristics on different sample

Sample	Entropy	Difference from 8	Horizontal Correlation coefficient	Vertical Correlation coefficient	Diagonal Correlation Coefficient	NPCR	Difference From ideal NPCR value	UACI	Difference From ideal UACI value
1	7.9665	0.0335	0.0089	0.0115	0.011	98.8109	0.7985	33.1373	0.5062
2	7.9691	0.0309	0.0092	0.0076	0.009	99.3872	0.2222	32.7532	0.8903
3	7.9706	0.0294	0.012	0.0116	0.009	98.9975	0.6119	32.7139	0.9296
4	7.9654	0.0346	0.0062	0.0135	0.007	99.5617	0.0477	32.8416	0.8019
5	7.9859	0.0141	0.007	0.0068	0.0133	99.2196	0.3898	32.9089	0.7346
6	7.9726	0.0274	0.006	0.0107	0.0125	99.1055	0.5039	33.069	0.5745
7	7.9798	0.0202	0.01	0.0109	0.0062	98.9286	0.6808	33.1407	0.5028
8	7.966	0.034	0.012	0.0129	0.0098	99.5583	0.0511	32.8987	0.7448
9	7.9782	0.0218	0.0089	0.0122	0.0118	98.8209	0.7885	32.9629	0.6806
10	7.9826	0.0174	0.0113	0.0105	0.0071	99.6167	0.0073	33.2033	0.4402
11	7.9675	0.0325	0.0083	0.0118	0.0068	98.8386	0.7708	32.8391	0.8044
12	7.9724	0.0276	0.0108	0.0128	0.0096	99.6075	0.0019	33.1326	0.5109
13	7.9786	0.0214	0.0062	0.0098	0.0115	99.3388	0.2706	33.2944	0.3491
14	7.9718	0.0282	0.011	0.0056	0.0129	99.02	0.5894	33.2725	0.371
15	7.9774	0.0226	0.0057	0.0066	0.0114	99.0794	0.53	33.0107	0.6328
16	7.9755	0.0245	0.0119	0.0093	0.0069	99.1211	0.4883	33.4402	0.2033
17	7.9848	0.0152	0.0061	0.0088	0.0057	99.3127	0.2967	33.3064	0.3371
18	7.9795	0.0205	0.0078	0.0066	0.0069	98.9378	0.6716	33.1887	0.4548
19	7.9889	0.0111	0.0093	0.0104	0.0131	99.6442	0.0348	33.3259	0.3176
20	7.9826	0.0174	0.0111	0.0087	0.0128	99.3223	0.2871	33.4022	0.2413
21	7.9672	0.0328	0.0069	0.0129	0.0131	98.9333	0.6761	32.9007	0.7428
22	7.9664	0.0336	0.0069	0.0057	0.0089	99.118	0.4914	33.363	0.2805
23	7.9686	0.0314	0.006	0.0072	0.0127	99.5734	0.036	32.886	0.7575
24	7.9737	0.0263	0.0068	0.0057	0.0064	99.1019	0.5075	33.0008	0.6427
25	7.9728	0.0272	0.0082	0.0066	0.0074	99.1494	0.46	33.162	0.4815
26	7.9808	0.0192	0.0132	0.0073	0.0073	99.4602	0.1492	33.1601	0.4834
27	7.9834	0.0166	0.0125	0.0084	0.0073	99.1357	0.4737	33.4025	0.241
28	7.9834	0.0166	0.013	0.007	0.0069	99.06	0.5494	33.0275	0.616
29	7.9768	0.0232	0.0122	0.0069	0.0083	99.3253	0.2841	33.3724	0.2711
30	7.9713	0.0287	0.0059	0.007	0.0094	99.0799	0.5295	33.0671	0.5764

- Step 1:- To compute the correlation matrix .
 Step 2:- To calculate the eigen value and eigen vectors from correlation matrix.
 Step 3:- To compute the factors as

$$\sqrt{\text{eigen value}} \times (\text{eigen vector})^2$$

- Step 4:- To add the cumulative effect value for all these data items.
 Using Factor analysis on the above data set the Encryption Factor (Ef) determines as

$$Ef = \text{abs}(8 - \text{entropy}_{val}) \times .956 + hco \times .9762 + vco \times .9780 + dco \times .9904 + \text{abs}(\text{abs}(66.6094 - Nval) \times .9945 + \text{abs}(33.6435 - Uval) \times .9528;; \quad (30)$$

//hco=horizontal correlation coefficient;; vco= vertical correlation coefficient;; dco=diagonal correlation coefficient;;
 Nval= NPCR value ,Uval=UACI value;;;

Example

The encryption factor of following outputs is given below in Table 2.

Table 2. Computation of encrypted factor

Sl no	Entropy	hoc	vco	dco	NPCR	UACI	Encryption Factor
31	7.9714	-0.147	0.016	0.0104	98.8427	33.4875	0.8209
32	7.9875	0.0148	0.0114	0.0062	99.544	33.2574	0.4766

Then Selection() process is applied to select the best encrypted image. This method is persisted up to a specified number of generations. Finally initial values for which best encrypted image is obtained are selected as optimized key stream. Last four bits of secret key is denoted as x_6 embedded with this optimal key to make the 64 hexadecimal numbers decryption key.

Selection(Encrypted Image(EI),Initial Population(IP),Maximum no.of Generation(maxgen))

```

1. set Ef = 1; ; maxgen = 5;
2. Generate initial population (IP); Key = IP //Initial population is generated from sec
4.1 and from sec 4.2.Among 64 bits hexadecimal number 60 bits are used as initial value of chaotic equation and 4 bits
are used to produce a number. This number is used as number of iterations of the chaotic system for removing its adverse
effects.
3. Create encrypted image (EI)
4. etr = entropy(EI);
   n = npcr(EI, img) ; //img stands for input image
u = UACI(EI, img)
   vcc = vertical_CC (EI);
hcc = horizontal_CC (EI);
   dcc = diagonal_CC(EI) // CC stands for correlation coefficient
5. while i<maxgen
   Produce IP' using crossover and mutation of IP
   Produce encrypted image using EI' using IP'
   etr' = entropy(EI');
   n' = npcr(EI, img') ; //lmg stands for input image
   u' = UACI(EI, img')
   vcc' = verticalCC (EI');
   hcc' = horizontalCC (EI');
   dcc' = diagonalcc(EI');
   Ef' = Encryption factor(etr', vcc', hcc', dcc', n', u');
   if (Ef' < Ef) then Ef = Ef1
End of if
   IP = IP' ;i=i+1
End of while.
6. return key;

Optimized key = (key + x6) = 256 bit key stream (31)
    
```


4.5. Decryption

Decryption process is just reverse of encryption process. Just follow the sub functions of our proposed algorithm in reverse order that we perform in encryption for getting original image back. Decryption process started with optimized 256-bit number. Initial key generation as stated in sec 4.1 is applied for gaining initial values of chaotic system. Chaotic system is solved and five random sequences are generated for features processing. First, we perform bitwise XOR operation then circularly rotate each pixel value in reverse order. Next positions of the pixels are shuffled according sec 4.3 in reverse order. Finally original input image is obtained.

5. Results and Discussion

We implement the proposed algorithm on different medical images from ‘Kaggle.com.’ Various images of size 256×256 are taken as plain images. A grayscale medical image of (256×256) is chosen for stepwise demonstration of our proposed algorithm. The first image is processed according to the secret key generation algorithm described in sec 4.1. Then, an encrypted image is generated. A genetic algorithm is applied to optimize the initial values of hyper-chaotic equations. Lastly, the best initial values of a chaotic system are obtained. Analysing the results, it is concluded that the 4th generation gives the better encryption result. So, the initial bit stream for the 4th generation is selected as the optimized key. A step-by-step detailed outcome of the proposed algorithm is given in Table. 3.

Table 3. A stepwise result demonstration

Secret Key value	Generations	Breakup value from secret key		Initial value for Crypto System	Apply proposed Encryption Algorithm	Selection process()	Initial value for best encrypted image	Key value for Decryption
8d6668709d4382de39edc51d88ce33208525aa deed45e051cea60cbfa1e479b6	Initial Population	x1	8d6668709d43	0.015547098	Encrypted Image	Select Best Encrypted Image		8d663ded212582de8c202d5188cee84545e4adeb9bf0543cea6b370601d79b6
		x2	82de39edc51d	0.014389097				
		x3	88ce33208525	0.01504192				
		x4	aadeed45e051	0.018787444				
		x5	cea60cbfa1e4	0.022721257				
	1st Generation	x1	8d66b92060e4	0.015547233	Encrypted Image			
		x2	82deb3452143	0.0143893				
		x3	88ce6dbf2d1d	0.015042019				
		x4	aade8c704525	0.018787282				
		x5	cea6e8ed0551	0.022721627				
	2nd Generation	x1	8d6633bfc551	0.015547009	Encrypted Image			
		x2	82deed7085e4	0.014389398				
		x3	88ce0cede043	0.015041856				
		x4	aade6820a11d	0.018787221				
		x5	cea63945ad25	0.022721332				
	3rd Generation	x1	8d663ded2125	0.015547027	Encrypted Image		8d663ded2125	
		x2	82de8c202d51	0.014389235			82de8c202d51	
		x3	88cee84545e4	0.015042224			88cee84545e4	
		x4	aadeb9bf0543	0.018787358			aadeb9bf0543	
		x5	cea6b370601d	0.022721537			cea6b370601d	
4th Generation	x1	8d660c45851d	0.015546943	Encrypted Image				
	x2	82de68bfe025	0.014389175					
	x3	88ce3970a151	0.015041931					
	x4	aade33edade4	0.018787133					
	x5	cea6bd20c543	0.022721553					

5.1. Statistical Analysis

A. Histogram Analysis

The distribution of an image's pixel value level is displayed in a histogram. An image with a disproportional distribution is open to incursion. An encryption algorithm should be able to create a cipher text image with a level distribution that is robust to statistical attacks. The chi-square test allows us to make sure uniformity, the chi square value is calculated by the equation described below [23].

$$\chi^2 = \sum_{i=0}^{N_p-1} \frac{(o_i - e_i)^2}{e_i} \quad (32)$$

For a particular grey value, the observed and expected instances are denoted by o_i and e_i respectively. The ideal chi-square value (χ^2) at significance 0.05 is 293.25. In Table 4 we represent the histograms of normal images and plain images of different samples. From the outcome, it is observed that all the histograms of encrypted images are almost the same and uniform at a certain level. Moreover, the calculated chi-square values for different samples are showed in Table 5.

B. Correlation Coefficient of Adjacent Pixels

The link between two variables is measured by the correlation coefficient $r(x,y)$. The correlation coefficient is useful for determining the similarity between two variables. The value of the correlation coefficient is always in the range between -1 and 1. Two variables are positively linked if the value of $r(x,y)$ is larger than 0, and negatively correlated if the value of $r(x,y)$ is less than 0. Two variables are substantially independent if the value is 0, or near to zero.

This test is very important one to determine the efficiency of the encryption algorithm. Two identical images prove they are perfectly correlated and their correlation coefficient is 1. Similarly, the correlation coefficient of completely different image is very closer to 0. Hence it can be concluded that the correlation coefficient between plain image and its associated encrypted image is very low means closer to 0 for a good encryption algorithm[24]

Let two-pixel values of the same location of plain text image (original image) and its produced encrypted image be denoted as x and y , then the formula of the correlation coefficient is given by the following:

$$r(x, y) = \frac{cov(x,y)}{\sigma_x \sigma_y} \quad (33)$$

$$cov(x, y) = \frac{1}{N} \sum (x_i - \bar{x})(y_i - \bar{y}) \quad (34)$$

$$\sigma_x = \sqrt{\frac{1}{N} \sum (x_i - \bar{x})^2} \quad (35)$$

$$\sigma_y = \sqrt{\frac{1}{N} \sum (y_i - \bar{y})^2} \quad (36)$$

N is the total number of pixels in the image.

In Table 6, the correlation values between the original images and their encrypted images are presented. The test findings confirm that the encrypted image shows a significantly lower correlation between adjacent pixels compared to the original image, which shows a high correlation. These results provide assurance regarding the effectiveness of our proposed algorithm. Moreover Figure 4 demonstrate the pixel distributions of the original and encrypted images in horizontal, vertical, and diagonal directions.

C. Information Entropy

The degree of randomness in an image is measured by information entropy. Information entropy can be calculated as:

$$I(m) = \sum_{i=0}^{M-1} P(m_i) \log_2 \frac{1}{P(m_i)} \quad (37)$$

Where, total number of pixel intensity value is represented by M and $P(m_i)$ is the probability of the occurrence of m_i . For a perfect system, the entropy value should be near to 8 [25]. The entropy indicates the probability distribution of the grey levels of through the image. The higher entropy value indicates the uniform distribution of the grey levels. So, pixels are distributed in such a way that it becomes harder for crypt analysts to understand even a small part of the image [26]. The entropy results of different input images furnished in table 6. It may be considered that proposed algorithm ensures a high entropy value and very close to the theoretical value. The level of information leakage is inconsiderable and it also find that proposed scheme can competently oppose entropy attack.

5.2. Differential Attacks

A highly effective image encryption algorithm is characterized by its ability to generate completely distinct cipher images despite very little alterations in input images, while using the same key and process. Researchers rely on two key metrics, the Number of Pixel Change Rate (NPCR) and the Unified Averaged Changed Intensity (UACI), to evaluate the algorithm's responsiveness to varying input images. These indicators can be formally expressed mathematically as follows

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100 \quad (38)$$

$$UACI = \frac{\sum_{i,j} |e(i,j) - e'(i,j)|}{255 * m * n} \times 100 \quad (39)$$

$$D(i, j) = \begin{cases} 0 & \text{when } e(i, j) = e'(i, j) \\ 1 & \text{when } e(i, j) \neq e'(i, j) \end{cases} \quad (40)$$

Where $m \times n$ is size of image. $e(i, j)$ and $e'(i, j)$ denotes two encrypted images of the original image and a slightly altered edition of the original image, respectively. In order to prevent vulnerability, it is important for the NPCR and UACI values to be high. According to V. According to Sangavi et al[25], the essential thresholds for NPCR and UACI are 99.5693% and 33.6447% respectively, set at a significance level of 0.05. Table 6 showcases the NPCR and UACI values acquired through the suggested algorithm. Findings from diverse test images illustrate that the proposed method outperforms the anticipated theoretical benchmark.

Analysis of Resisting Chosen -plaintext

There exist four primary categories of attacks within cryptanalysis: chosen plaintext attacks, known plaintext attacks, ciphertext-only attacks, and chosen cipher text attacks. It is suggested that if an encryption algorithm can withstand chosen plaintext attacks, it inherently becomes resilient against the other three types of attacks [27,28].

In chosen plaintext attacks, adversaries can freely select any plaintext for encryption, gaining its corresponding cipher text to extract maximum information. Our proposed encryption method addresses this by commencing the encryption process with a substantial number of user-chosen random values. Consequently, if all pixel values are uniform within a specific image, our cryptographic framework generates distinct ciphertext images in each instance. This robustness strengthens our encryption against chosen plaintext attacks.

Furthermore, our system harnesses a chaotic system to generate exceedingly random sequences employed at every stage of the encryption process. The response of a chaotic system heavily depends on its initial values, which, in our design, stem from a fusion of the input image and a sequence of user-selected random numbers. Hence, even a slight modification in the input image yields entirely different initial values for the chaotic sequence. This prevents adversaries from deciphering the chaotic sequences by analysing sets of plaintext and cipher text images, thereby nullifying preferred plaintext attacks.

In our cryptographic framework, a genetic algorithm is integrated to generate supplementary initial values within the chaotic system's starting parameters. These generated values are examined to select the one that yields an encrypted image with superior statistical parameter values, thereby forming the ultimate keyword. The amalgamation of initial values in our proposed system makes it exceedingly challenging for adversaries to extract any information from the encrypted image derived from the chosen plaintext.

5.3. Key Space Analysis

Chaotic sequence is an excellent feature is that a little change in the initial value produce a completely different scenario. To design a crypto system with enough reliability the key space is very important. Key space should be successful of neutralizing brute-force attacks. In our proposed system x_1, x_2, x_3, x_4, x_5 and x_6 are initial parameters for systems with chaotic features. The precision of the initial values should be as high as possible such that 14 to 15 digits after decimal point. The key space is $10^{70} \cong 2^{232} > 2^{100}$ so proposed algorithm can resist to brute force attack.

5.4. Sensitivity Analysis

An encryption algorithm should be very much sensitive to its encryption and decryption key [29-31]. It means if we change a single bit in key value during encryption it produces a completely different encrypted image as well as if we change a slight in input image it also produces different encrypted image. Similarly, if we change a single bit in key value during decryption then we cannot recover same encrypted image. To prove the sensitivity of our proposed algorithm let we take K as key value and derived a key K1 with one bit change. Described as follows

$$k = 8d6668709d4382de39edc51d88ce33208525aadeed45e051cea60cbfa1e479b6$$

$$k1 = 8d6668709d4382de39edc51d88ce33208525aadeed45e051cea60cbfa1e479b7$$

Fig. 3a represent the plain image. The encrypted image with k is shown in Fig 3b and encrypted image with k1 shown in Fig 3c .and the difference between Fig. 3b and Fig. 3c is shown Fig. 3d.

Our system states that optimized key value is used as the key for decryption. Let's for input image Fig.3a decrypted key be k_2 .and we change a single bit in k_2 to obtain k_3 . We apply k_2 and k_3 for decryption and the resultant decrypted image shown in Fig 3e and Fig 3f. Fig 3g describes the difference between Fig. 3e and Fig. 3f from the scenario it is concluded that if we change a single bit in decryption key it is impossible to retrieve the original image.

$$k2 = 8d663ded212582de8c202d5188cee84545e4aadeb9bf0543cea6b370601d79b6$$

$$k3 = 8d663ded212582de8c202d5188cee84545e4aadeb9bf0543cea6b371601d79b6$$

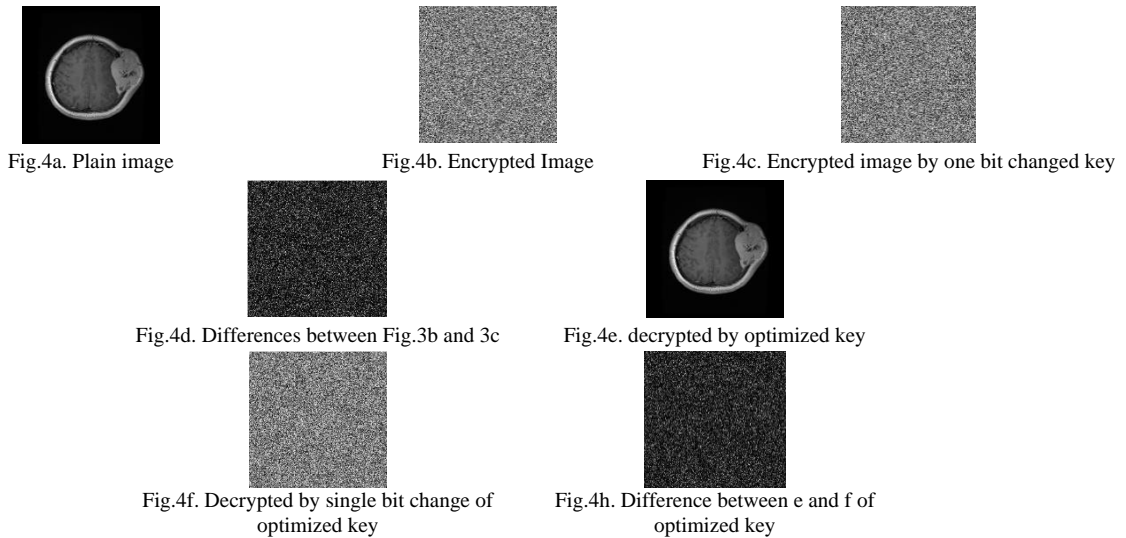


Fig.4.

5.5. Threat Model

A threat model is simply a generic list of threats that are considered common during the transmission of medical image. Transmission of medical image is very sensitive because a small change in encrypted or in decrypted image may cause wrong diagnosis. In this section a few attacks are considered. Since the transmission of medical image is done through unsecure public network as a result it may possible that

- an adversary makes an attempt to obtain the key value from public parameter
- an attacker attempted to make delete, alter and modify the encrypted image

An attacker may perform different analysis to get some hints about the transformation of input pixel to the output pixel.

In our proposed algorithm the 256 bit stream and the input image are the parameters of the keyword generation function. A small change in bit stream or in input image produce a completely different keyword so it is almost impossible for any adversary to rightly guess the key value.

Secondly if the attacker makes any modification in encrypted image, then proposed algorithm would produce a completely different decrypted image.

Finally, the histograms of the different encrypted images are almost flat this demonstrate that the frequency of all the pixel values in encrypted images are same. As a result, it is not possible to obtain any type of hint to establish the relation of input and output pixel value. The Experimental outcomes to defend different threat cases are described in Table 7 and in Table 8.

Table 4. Input image, encrypted image and their histograms

	Plain Image	Encrypted Image	Histogram of Plain Image	Histogram of Encrypted Image
Sample1				
Sample2				


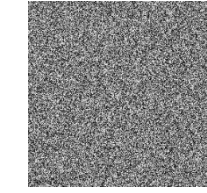
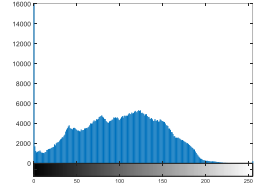
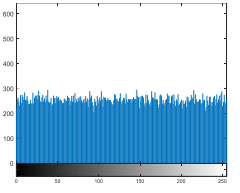

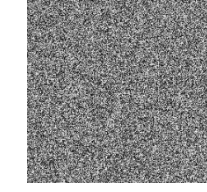
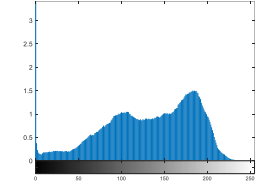
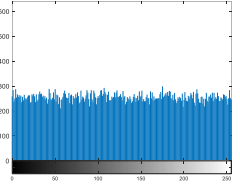
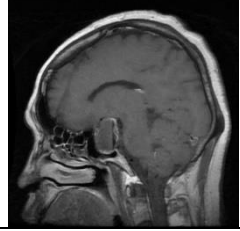
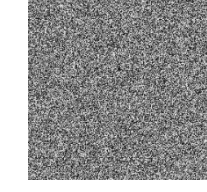
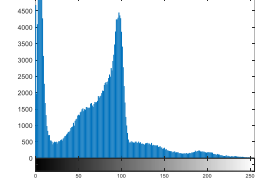
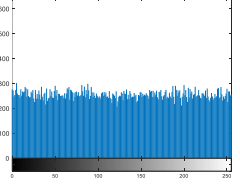

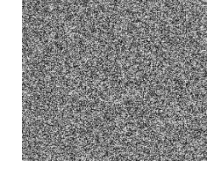
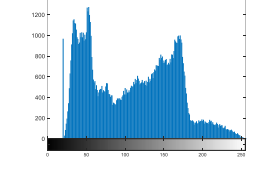
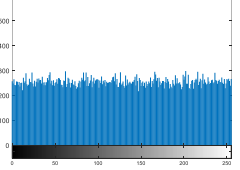

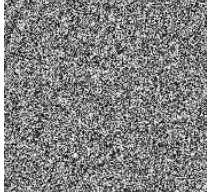
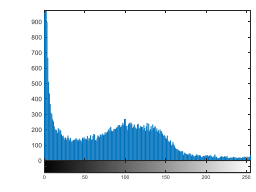
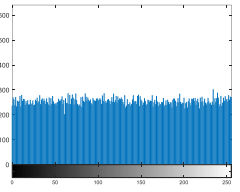
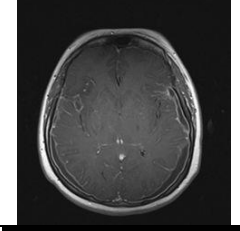
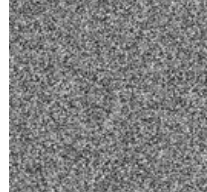
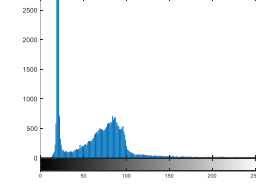
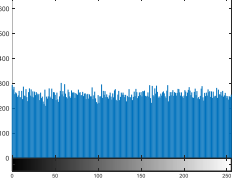
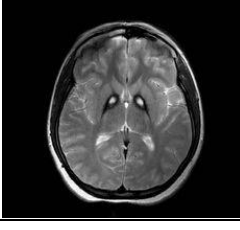
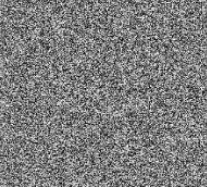
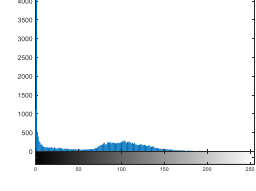
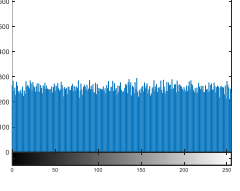
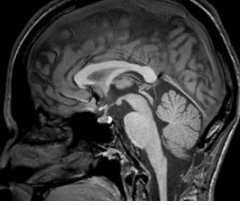
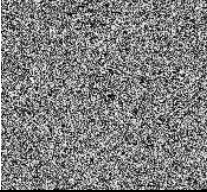
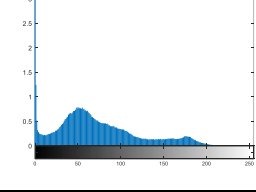
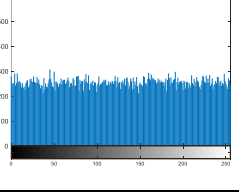
	Plain Image	Encrypted Image	Histogram of Plain Image	Histogram of Encrypted Image
Sample3				
Sample4				
Sample5				
Sample6				
Sample7				
Sample8				
Sample9				
Sample10				

Table 5. Chai square value of plain image and encrypted image

	Plain Image	Encrypted Image
Sample1	87587.8	254.65
Sample2	96412.7	258.64
Sample3	74559.4	269.3
Sample4	96543.32	256.01
Sample5	84077.2	264.12
Sample6	69248.87	257.18
Sample7	75254.32	258.32
Sample8	85321.06	262.32
Sample9	82472.36	259.21
Sample10	74824.24	277.4

Table 6. Output values of different tests

Plain Image	Correlation Coefficient						Entropy	UACI(%)	NPCR (%)
	Plain Image			Encrypted Image					
	horizontal	vertical	diagonal	horizontal	vertical	diagonal			
Sample1	0.96938	0.97268	0.94351	-0.00853	-0.00463	-0.00853	7.99862	33.42	99.635
Sample2	0.96904	0.96747	0.94054	-0.00260	-0.00254	-0.00521	7.99891	33.40	99.627
Sample3	0.96849	0.97333	0.93996	0.00702	-0.00602	-0.00736	7.99905	33.41	99.631
Sample4	0.99307	0.98796	0.98248	0.00144	0.00505	-0.00387	7.99915	33.38	99.597
Sample5	0.99306	0.98951	0.98378	-0.00185	-0.00112	-0.00109	7.99732	33.40	99.614
Sample6	0.96727	0.95831	0.92882	-0.00474	-0.00427	-0.00198	7.99894	33.39	99.626
Sample7	0.94351	0.93592	0.97361	0.00706	0.00547	0.00576	7.98305	33.37	99.574
Sample8	0.97561	0.93363	0.97608	0.00629	-0.00846	0.00875	7.97661	33.39	99.655
Sample9	0.96066	0.94853	0.95512	0.00748	-0.00654	0.00698	7.99757	33.38	99.487
Sample10	0.93053	0.94197	0.97361	0.00654	-0.00452	-0.00325	7.97436	33.38	99.558

Fig. 5a, Fig 5b, Fig 5c horizontal, vertical, and diagonal correlation distribution of plain image
 Fig. 5d, Fig. 5e, Fig. 5f horizontal, vertical, and diagonal correlation distribution of encrypted image

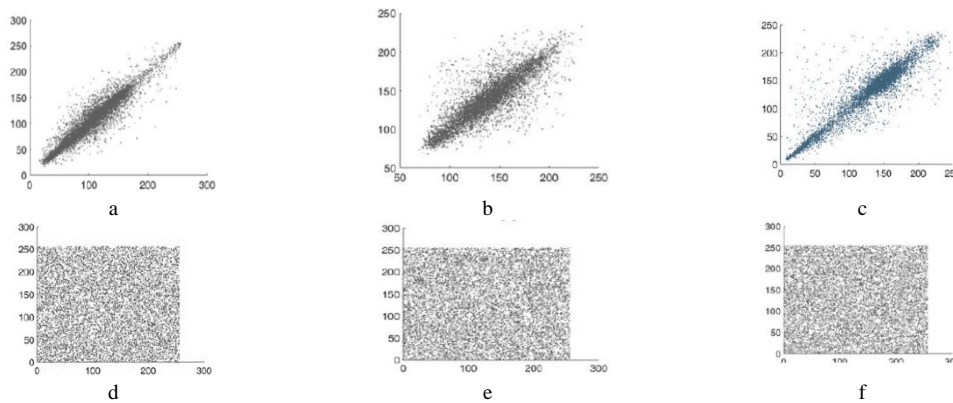


Fig.5. Scatter plot of adjacent pixel correlation a) horizontal b) vertical c) diagonal direction of input image d) horizontal e) vertical f) diagonal direction of encrypted image

6. Comparison

We compare the results that were generated from our proposed system with algorithm defined in [32, 33,34 and 35] with respect to the correlation coefficient, entropy values. For proper comparison we execute different algorithms on same set of samples. The correlations of pixels of encrypted image in different directions is very close to other algorithms. Entropy values of our proposed algorithm is very close to the ideal value like other algorithms. Table 9 describes details result of comparisons. The result demonstrates the efficiency of proposed method.

Table 7. Generation of key by small changing of random value and input image

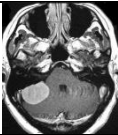
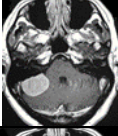
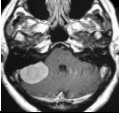
Chosen random value	Input image	Generated Key value	Remarks
2b5fd4435f3f95cf 79dd9bd3704912aa 2aacf977ba79e023 d65ceac0649f8cb2		d7e8cb8bafea54d2 0cb70322c91a112c 27272ac85c02bcd0 80e2986c5fd7d784	Any random value and input image generates key value
2b5fd4435f3f95cf 79dd9bd3704912aa 2aacf977ba79e023 d65ceac0649f8cb3		9712f3e5f9648634 ce4f5ae5a5ab7be6 910cd0ebba55a853 fe8c8e558b08d867	If we change the last bit of chosen random value proposed scheme will generate a different key value
2b5fd4435f3f95cf 79dd9bd3704912aa 2aacf977ba79e023 d65ceac0649f8cb2		cd492dfc251d1faa 320f7f3e21c52e2c 27697141682ee23c 737c05430746f409	If we change a single bit of the input image with same random value it also generates different key value

Table 8. Generation of decrypted image if encrypted image is modified

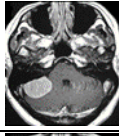
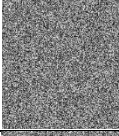
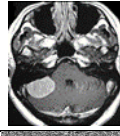

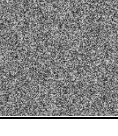
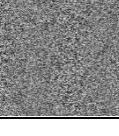
Input image	Key Value	Encrypted Image	Decrypted Image	Remarks
	9e7bf092cbe0d1f7 4a00687b2bb90dd5 fff204d5573681d1 d0bed889ba1ae596			Actual key value and encrypted image produce decrypted image same as input image.
	9e7bf092cbe0d1f7 4a00687b2bb90dd5 fff204d5573681d1 d0bed889ba1ae596			If we change the encrypted image a little then it generates a different image though the key value is same

Table 9. Comparisons results of proposed algorithm with other algorithms

Method	Image	Correlation coefficient			Entropy
		horizontal	vertical	diagonal	
Proposed	Sample 1	-0.00853	-0.00463	-0.00853	7.99862
	Sample 2	-0.00260	-0.00254	-0.00521	7.99891
Ref [32]	Sample 1	0.00214	-0.00486	-.00314	7.99851
	Sample 2	0.00315	0.00287	.00264	7.99884
Ref [33]	Sample 1	0.00198	-0.00484	0.0016	7.99894
	Sample 2	0.0037	0.00212	0.0005	7.99774
Ref [34]	Sample 1	0.00125	0.00214	0.00147	7.98457
	Sample 2	0.00224	0.00325	0.0011	7.99547
Ref [35]	Sample1	0.00321	0.00452	0.00331	7.98457
	Sample2	0.00351	0.00341	0.0042	7.97845

7. Conclusions

In this paper, we propose a chaos-based symmetric medical image encryption. The proposed method generates encrypted images and handles the challenge of selecting the correct initialization for the hyper-chaotic system. This enhancement involves refining the initial values of the hyper-chaotic system using a genetic algorithm. Initially, a robust key value is derived from the input image to thwart chosen and known plaintext attacks. Subsequently, leveraging these key values, the encryption algorithm progresses through two intricate and convoluted stages. In the subsequent stage, a genetic algorithm is employed to determine the most suitable initial values for the chaotic equations. Although this optimization technique slightly increases the computational time of our suggested algorithm, it significantly heightens its security level. The outperforms of the proposed algorithm are satisfactory in terms of NPCR, entropy, UACI, and Chi-square value by 99.62%, 7.995, 33.4%, and nearly 270, respectively. The outcomes of these tests have validated that the proposed scheme exhibits superior security and remarkable efficiency in encrypting medical images. Our future endeavours will aim to streamline the computational time involved in the process.

References

- [1] S. Ibrahim, Hesham Alhumyani, Mehedi Masud, Sultan S. Alshamrani, Omar Cheikhrouhou, Ghulam Muhammad, M. Shamim Hossain “Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps”, *IEEE Access*, volume 8, pages 160433-160449,2020.
- [2] Zhongyun Hua, Shuang Yi, Yicong Zhou, “Medical image encryption using high-speed scrambling and pixel adaptive diffusion”, *Signal Processing*, volume 20, pp 134–144 ,2018.
- [3] Weijia Cao, Yicong Zhou, C.L. Philip Chen, Liming Xia, “Medical image encryption using edge maps”. *Signal Processing*. Volume 132, pp 96–109,2017.
- [4] Chong Fu, Gao-yuan Zhang, Ou Bian, Wei-min Lei, Hong-feng Ma, “A novel medical image protection scheme using a 3-dimensional chaotic system”, *PLoS One*, volume 9, pp 1–25, 2014.
- [5] Guodong Ye, Kaixin Jiao, Chen Pan, Xiaoling Huang, “An effective framework for chaotic image encryption based on 3D logistic map”, *Security and Communication Networks*, Volume2018, pp 1–11,2018.
- [6] Y. Dai, X. Wang, “Medical image encryption based on a composition of Logistic maps and Chebyshev maps, 2012 IEEE International Conference on Information and Automation, Shenyang, China, pp 210-214 ,2012.
- [7] Chong Fu, Wei-hong Meng, Yong-feng Zhan, Zhi-liang Zhu, Francis C.M.Lau ,Chi K.Tse ,Hong-feng Ma , “An efficient and secure medical image protection scheme based on chaotic maps”, *Computers in Biology and Medicine* volume 43,issue 8 pages 1000–1010, 2013. <https://doi.org/10.1016/j.combiomed.2013.05.005>
- [8] Jun-xin Chen, Lei Chen, Leo Yu Zhang, Zhi-liang Zhu, “Medical image cipher using hierarchical diffusion and non-sequential encryption”, *Nonlinear Dynamics* 96, pp 301–322. 2019, doi:10.1007/s11071-019-04791-3.
- [9] Yin Dai, Huanzhen Wang, Yuyi Wang, “Chaotic medical image encryption algorithm based on bit-plane decomposition”, *International Journal of Pattern Recognition and Artificial Intelligence* Volume 30, No. 4 1657001, 2016, <https://doi.org/10.1142/S0218001416570019>
- [10] Xiao Chen, Chun Jie Hu, “Adaptive medical image encryption algorithm based on multiple chaotic mapping”, *Saudi Journal of Biological Sciences*, Volume 24, Issue 8, Pages 1821-1827, 2017
- [11] Li Huijuan, Wang Yurong, Zuo Zhengwei, “Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms”, *Optics and Lasers in Engineering*, Volume 115, pp 197-207, 2019.
- [12] Shanshan Li, Li Zhao, Na Yang, "Medical Image Encryption Based on 2D Zigzag Confusion and Dynamic Diffusion", *Security and Communication Networks*, vol. 2021, Article ID 6624809, 23 pages, 2021. <https://doi.org/10.1155/2021/6624809>.
- [13] Javan, A.A.K.; Jafari, M.; Shoeibi, A.; Zare, A.; Khodatars, M.; Ghassemi, N.; Alizadehsani, R.; Gorriz, J.M. “Medical Images Encryption Based on Adaptive-Robust Multi-Mode Synchronization of Chen Hyper-Chaotic Systems”. *Sensors* 2021, 21, 3925
- [14] Liu, Shuang, Liu, Li, and Pang, Ming. “Encryption Method and Security Analysis of Medical Images Based on Stream Cipher Enhanced Logical Mapping” 1 pp 185 – 193, Jan. 2021.
- [15] Hui Wang, Di Xiao, Xin Chen, Hongyu Huang, “Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map”, *Signal Processing*, Volume 144, pp 444-452, 2018, <https://doi.org/10.1016/j.sigpro.2017.11.005>
- [16] Junxin Chen, Fangfang Han, Wei Qian, Yu-Dong Yao, Zhi-liang Zhu, “Cryptanalysis and improvement in an image encryption scheme using combination of the 1D chaotic map”, *Nonlinear Dynamics* 93,399–2413, 2018. doi:10.1007/s11071-018-4332-9.
- [17] Congxu Zhu, Guojun Wang, Kehui Sun, “Improved cryptanalysis and enhancements of an image encryption scheme using combined 1D chaotic maps”, *Entropy* 20 (11),843, 2018.
- [18] John H. Holland, *Adaptation in natural and artificial Systems*. “An introductory analysis with applications to biology, control, and artificial intelligence”, MIT Press, IEEE Explorer Online ISBN: 9780262275552.
- [19] A.E. Eiben, P-E. Raué, Zs. Ruttkay, “Genetic algorithms with multi-parent recombination. Proceedings of the 3rd conference on parallel problem solving from nature”, LNCS Series, Springer-Verlag, 1994
- [20] Amin Zarei, “Complex dynamics in a 5-D hyper-chaotic attractor with four-wing, one equilibrium and multiple chaotic attractors”, *Nonlinear Dynamics*. ISSN 0924-090X, Volume 81, Combined 1-2, *Nonlinear Dynamics* ,2015 81:585-605, DOI 10.1007/s11071-015-2013-5
- [21] Zhenjun Tang, Juan Song, Xianquan Zhang, Ronghai Sun, “Multiple-image encryption with bit-plane decomposition and chaotic maps, *Optics and Lasers in Engineering*, Volume 80, pp 1-11,2016, ISSN 0143-8166, <https://doi.org/10.1016/j.optlaseng.2015.12.004>.
- [22] Seyed Mohammad Seyedzadeh, S. Mirzakuchaki, “A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map”. Volume 92, Issue 5, pp 1202-1215,2012.
- [23] Hongjuan Liu, Zhiliang Zhu, Huiyan Jiang, Beilei Wang, “A novel image encryption algorithm based on improved 3D chaotic cat map”, *The 9th International Conference for Young Computer Scientists, Northeastern University, Shenyang, Liaoning, China, ICYCS 2008*. 3016-3021. 10.1109/ICCIC.2010.5705910. 2008.
- [24] Shun Zhang, Tiegang Gao, an image encryption scheme based on DNA coding, and permutation of hyper-image, *Multimedia Tools and Applications*. 75, pp 17157– 14170.201, doi: 10.1007/s11042-015-2982-
- [25] V.Sangavi, P.Thangavel, “An exotic multi-dimensional conceptualization for medical image encryption exerting rossler system and Sine map”, *Journal of Information Security and Applications*, Volume 55,102626,ISSN 2214-2126,2020
- [26] Manjit Kaur, Dilbag Singh, “Multi objective evolutionary optimization techniques based hyper chaotic map and their applications in image encryption”, *Multidimensional Systems and Signal Processing* 32(1), <https://doi.org/10.1007/s11045-020-00739-8> ,2020
- [27] Xiaodong Li, Cailan Zhou, Ning Xu, “A secure and efficient image encryption algorithm based on DNA coding and Spatiotemporal chaos”, *International Journal of Network Security* , Volume 20, No 1, pp 110-120, 2018
- [28] Changjiang Zhu, Zhihua Gan, Yang Lu, Xiuli Chai, “An image encryption algorithm based on 3-D DNA level permutation and substitution scheme”, *Multimedia Tools and Applications* 79(4), pp 7227–7258, 2020. <https://doi.org/10.1007/s11042-019-08226-4>

- [29] Huijuan Li, Yurong Wang, Zheng-Wei Zuo, “Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms”, *Optics and Lasers in Engineering*, volume 115, pp 197–207,2019
- [30] Hossein Nematzadeh, Rasul Enayatifar, Homayun Motameni, Frederico Gadelha Guimarães, Vitor Nazário Coelho, “Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices”, *Optics and Lasers in Engineering*, volume 110, pages 24–32,2018
- [31] Suresh N.Mali,Pradeep M.Patil ,Rajesh M.Jalnekar, “Robust and secured image-adaptive data hiding”, *Digital Signal Processing*. Volume 22, issue 2 pp 314–323. 2012
- [32] Prema T, Akkasaligar, Sumangala Biradar, “Selective medical image encryption using DNA cryptography”. *Information Security Journal,A Global Perspective*. Volume 29, issue 2, pp 91-101,2020 10.1080/19393555.2020.1718248
- [33] Joshua C. Dagadu, Jianping Li, Emelia O, Aboagye, Faith K Deynu, “Medical Image Encryption Scheme Based on Multiple Chaos and DNA coding”, *International Journal of Network Security*, Volume 21, No 1, pp 83-90,2019
- [34] Belazi, Akram, Muhammad Talha, Sofiane Kharbech, Wei Xiang, “Novel medical image encryption scheme based on chaos, and DNA encoding”. *IEEE Access*. 7, pp 36667– 36681,2019 doi: 10.1109/ACCESS.2019.2906292
- [35] Kiran, Parameshchhari, “Selective image encryption of medical images based on threshold entropy and Arnold Cat map”, *Bioscience Biotechnology Research Communications*. Vol 13. pp 194-202,2020

Authors' Profiles



Subhajit Das is currently a research scholar in Kalyani University, He Received M.Sc in Computer Science and MBA in information system. His research interest includes network security, digital image processing, medical image encryption. He has published more than 15 conference and journal papers.



Manas K. Sanyal is a Professor in the Department of Business Administration, University of Kalyani, India. He received his M. Tech degree and Ph.D. in Information Technology. Professor Sanyal has published several research papers in international journals of repute and co-authored a number of books. His interest includes Big Data, Machine Learning, Deep Learning, Business Intelligence and Cloud Computing.

How to cite this paper: Subhajit Das, Manas Kumar Sanyal, "A Hyper-chaotic Medical Image Encryption with Optimized Key Value", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.16, No.3, pp.18-34, 2024. DOI:10.5815/ijisa.2024.03.02