KSÜ Mühendislik Bilimleri Dergisi, 26(1), 2023
Araştırma Makalesi

KSU J Eng Sci, 26(1), 2023
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 22.11.2022
Kabul Tarihi : 25.01.2023

Received Date : 22.11.2022
Accepted Date : 25.01.2023

A NEW IMAGE ENCRYPTION METHOD BASED ON A 6D HYPERCHAOTIC MAP AND GENETIC OPERATORS

HİPERKAOTİK HARİTA VE GENETİK OPERATÖRLERE DAYALI YENİ BİR GÖRÜNTÜ ŞİFRELEME YÖNTEMİ

Mehmet DEMİRTAŞ[1*] (ORCID: 0000-0002-9018-3124)

[1] Necmettin Erbakan Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü, Konya, Türkiye

*Sorumlu Yazar / Corresponding Author: Mehmet DEMİRTAŞ, mdemirtas@erbakan.edu.tr

ABSTRACT

This paper presents a novel and secure image encryption method. The plain image's pixels are confused using the N-point crossover operation of genetic algorithms. Randomly paired rows and columns are determined by the two state variables of a six-dimensional hyperchaotic map. The number of crossover points, which are calculated by the two other state variables of the hyperchaotic map, differ from each other for each row or column pair. The crossover positions are specified according to the number of crossover points with the help of the last two state variables. The proposed algorithm generates the diffusion stage's encryption key using the SHA-256 hash value of the plain image. Mutation and crossover operators are implemented using the 16-bit subblocks of the 256-bit hash value. The scrambled image's pixels are altered with the generated encryption key and previously encrypted pixels. Keyspace and sensitivity, histogram, correlation, information entropy, differential, data loss, noise attack, and computational time analyzes are performed to test the safety and effectiveness of the encryption method. The experiments and simulation results show that the proposed encryption technique is highly secure and efficient since it can resist various attacks.

Keywords: chaos, crossover, genetic operations, hyperchaos, image encryption

ÖZET

Bu makale, yeni ve güvenli bir görüntü şifreleme yöntemi sunmaktadır. Düz görüntünün pikselleri, genetik algoritmaların N noktalı çaprazlama işlemi kullanılarak karıştırılır. Rastgele eşleştirilmiş satırlar ve sütunlar, altı boyutlu bir hiper kaotik haritanın iki durum değişkeni tarafından belirlenir. Hiperkaotik haritanın diğer iki durum değişkeni tarafından hesaplanan geçiş noktalarının sayısı, her satır veya sütun çifti için birbirinden farklıdır. Geçiş konumları, son iki durum değişkeni yardımıyla geçiş noktalarının sayısına göre belirlenir. Önerilen algoritma, düz görüntünün SHA-256 hash değerini kullanarak difüzyon aşamasının şifreleme anahtarını üretir. Mutasyon ve çaprazlama operatörleri, 256 bitlik hash değerinin 16 bitlik alt blokları kullanılarak gerçekleştirilir. Karıştırılan görüntünün pikselleri, oluşturulan şifreleme anahtarı ve önceden şifrelenmiş piksellerle değiştirilir. Şifreleme yönteminin güvenliğini ve etkinliğini test etmek için anahtar alanı ve duyarlılığı, histogram, korelasyon, bilgi entropisi, diferansiyel, veri kaybı, gürültü saldırısı ve hesaplama süresi analizleri yapılır. Deneyler ve simülasyon sonuçları, önerilen şifreleme tekniğinin çeşitli saldırılara karşı koyabilmesi nedeniyle oldukça güvenli ve verimli olduğunu göstermektedir.

Anahtar Kelimeler: çaprazlama, genetik işlemler, hiperkaos, kaos, görüntü şifreleme

ToCite: DEMİRTAŞ, M., (2023). A NEW IMAGE ENCRYPTION METHOD BASED ON A 6D HYPERCHAOTIC MAP AND GENETIC OPERATORS. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 26(1),261-278.

INTRODUCTION

A secure transfer of data between the sender and the receiver is of great importance in today's world of digitization. Digital images, as a type of digital data, are extensively used in people's daily lives due to advancements in information technology. For example, a great number of digital images are shared between people who use social networks. However, a confidential and private image could be intercepted, tampered with, and duplicated during the process of transmission over public networks. It is a significant task to develop ways for protecting the privacy of digital images. One of the methods to ensure secure image transmission is cryptography. Cryptosystems aim to make the plain image unrecognized by an intruder who can access the transmission channel.

Chaotic maps can be used in the architecture of cryptosystems for image encryption since they are extremely sensitive to initial conditions, unpredictable and non-periodic (Muthu and Murali, 2021). On the other hand, hyperchaotic maps, which contain more than one positive Lyapunov exponent, have more complex chaotic behavior than chaotic maps (Boriga, Dăscălescu, and Priescu, 2014). Hyperchaotic systems can also generate several random sequences with larger key space so that the encryption algorithm could be designed more securely (Q. Zhang and Han, 2021). Hyperchaotic systems can be constructed by modifying traditional chaotic maps. A new hyperchaotic Lorenz system was proposed by (Jia, 2007) by adding a state feedback controller to the well-known Lorenz system. Similarly, a hyperchaotic 4D Chua system was created (Xi, Yu, Zhang, Deng, and Xi, 2010) in which a state feedback controller was introduced to the second and third equation of the conventional 3D Chua system. A 5D conservative hyperchaotic system was offered by Dong et.al (2019). This hyperchaotic system, which has two positive Lyapunov exponents, was obtained by adding an extra function to a 4D chaotic system. In (Wang Fa-Qiang, 2006), an extra state variable was added to the Liu chaotic system to form a new 4D hyperchaotic system. Another novel multi-wing hyperchaotic attractor was proposed by Grassi et.al (2009). The system couples two identical Lorenz systems to generate a 6D hyperchaotic system.

Similar to chaotic maps, hyperchaotic maps are utilized in image encryption techniques (Demirtaş, 2022; Kaur and Kumar, 2020). Cao et.al (2018) created a novel 2D hyperchaotic map and used the map to implement a bit-level encryption method. Similarly, a 2D hyperchaotic map, which combines a sine map with a Henon map, was analyzed (Natiq, Al-Saidi, Said, and Kilicman, 2018) and this map is used to confuse and diffuse a plain image. Kaur et.al (2020) produced the secret keys of an encryption algorithm with larger key space using a 7D hyperchaotic map. In the study (Luo, Zhou, Liu, Cao, and Ding, 2018), a 4D hyperchaotic system was used to create an encryption matrix and permutation sequences. Pseudo-random sequences generated by a 6D hyperchaotic map were used to rotate the bit-plane matrix (Xu, Sun, and Wang, 2020) for a grayscale image encryption algorithm. A bit level-permutation process was implemented using the chaotic sequences generated by a hyperchaotic sequence with the help of the SHA-256 hash value of the plain image (Patro, Acharya, and Nath, 2019). Chen et.al (2020) suggested a new method to obtain a pseudorandom number generator (PRNG) using four-wing hyperchaotic systems and they used the designed PRNG in an image encryption method. A color image encryption algorithm was implemented by Cheng, Wang, Chen, and Chaos (2019) by generating the key streams of the diffusion phase using a 5D multi-wing hyperchaotic map. A 7D hyperchaotic map's initial values were generated by the SHA-512 hash function and the calculated chaotic sequences were used in the confusion and diffusion operations in (Sun, Guo, and Wu, 2019). Zhu and Zhu (2019) created a new discrete 5D hyperchaotic map by combining a logistic map and a discrete Lorenz map. The formed map was used in a block-based image encryption scheme. A 5D conservative hyperchaotic system (Zhou and Wang, 2020) was used both in the permutation and substitution phases of a novel image encryption method. Hyperchaos is not only used in the confusion-diffusion image encryption scheme but also have been combined with other techniques such as DNA encoding (Hui, Liu, and Fang, 2021; Mohamed, ElKamchouchi, and Moussa, 2020; T. Wang and Wang, 2020; Wu, Shi, and Li, 2020), cellular automata (Yaghouti Niyat, Moattar, and Niazi Torshiz, 2017; Zeng and Wang, 2021), and fractional calculus (Gao, Yu, Banerjee, Yan, and Mou, 2021; P. Li, Xu, Mou, and Yang, 2019; Yang, Mou, Liu, Ma, and Yan, 2020).

Crossover and mutation are examples of genetic operators that are inspired by biological evolution mechanisms (Katoch, Chauhan, and Kumar, 2021). These operators can be used in the image encryption processes. For example, a selection-crossover-mutation-based image encryption method was proposed (Xingyuan Wang and Xu, 2014). Guesmi et.al (2016) used a two-point crossover operation to shuffle the rows and columns of a color image's R, G, and B channels. In (Mozaffari, 2018), single-point crossover and flipping mutation operations were used for permutation and substitution of the plain image's pixels. Chai et.al (2021) suggested a novel encryption method in which DNA encoded sequences were permuted by double crossover operation and diffused by mutation operation.

In the work (Y.-Q. Zhang, He, Li, and Wang, 2020), Different types of mutations and two-point crossover operations were used to encode a plain image's pixels at the bit level. Niu et.al (2020) used selection, crossover, and mutation operators along with DNA encoding and Henon map to scramble and diffuse pixels. Gupta et.al (2021a) employed single-point and two-point crossover with mutation operator to obtain a session key which is used to encrypt a plain image. In a similar study (Gupta, Gupta, and Shukla, 2021b), a session key was generated with the help of a chaotic map and a two-point crossover operation.

This paper proposes a novel encryption method based on a 6D hyperchaotic map and genetic operators such as N-point crossover and mutation. The plain image is shuffled twice with the parameters generated by the 6D hyperchaotic map using the N-point crossover operation. The crossover operation creates new offspring by exchanging the genes of the parents. The plain image's rows and columns can be considered parents. The pixels of the plain image represent genes. The rows and columns of the plain image are paired with each other pseudo-randomly with the help of the chaotic sequences generated by the two state variables of the hyperchaotic map. The number of crossover points and the positions of the crossover points are also determined by the state variables of the 6D hyperchaotic map for each row or column pair. In the second round of permutation, the initial values of the hyperchaotic map are updated. Consequently, the row and column pairs, the number of crossover points, and the positions of the crossover points are recalculated. A 256-bit secret key is produced by using the SHA-256 hash algorithm from the plain image for the diffusion stage. The secret key is divided into 16 subblocks consisting of 16 bits. A mutation operation is implemented on each subblock by flipping one bit. In addition, a two-point crossover operation is performed between different subblocks to further scramble the secret key. The modified secret key depends on the plain image which makes the encryption method resistive against chosen-plaintext and chosen-ciphertext attacks. This secret key is used in the diffusion process which alters the values of the pixels of the permuted image. The image encryption algorithm can withstand statistical and differential attacks with the help of the diffusion process. To sum up, a novel and safe image encryption architecture which can effectively resist several attacks is presented in this work.

The remaining of this paper is planned as follows: Section 2 gives details of the methods used in the image encryption algorithm. Section 3 explains the proposed cryptosystem. Section 4 presents various simulation results of the proposed method. In Section 5, the conclusion part summarizes the overall work.

MATERIALS AND METHODS

6D Hyperchaotic Map

The 6D hyperchaotic system (Grassi et al., 2009) used in the scrambling operation of the proposed algorithm can be expressed as follows:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = bx - y - xz + \sigma_1(w - u) \\ \dot{z} = xy - cz \\ \dot{w} = a(u - w) \\ \dot{u} = bw - u - vw + \sigma_2(x - y) \\ \dot{v} = wu - cv \end{cases} \tag{1}$$

where x, y, z, w, u, and v are the state variables, and a, b, c are the system parameters and σ_1 and σ_2 are the scaling factors of the linear coupling terms. When $a = 10$, $b = 28$, $c = 8/3$, $\sigma_1 = 0.1$, $\sigma_2 = 0.01$ are chosen as the parameters for the system given in Eq. (1), then the four-wing attractors of the (x, w, z) and (x, u, v) spaces can be obtained as shown in Fig. 1, respectively.

The Shuffling Process by N-Point Crossover Operation

As a well-known genetic operator, the crossover operation essentially exchanges genetic information between parents to generate new offspring. N-point crossover operation requires N different crossover points so that a total of N+1 segments can be obtained for each parent. The genetic information in between the crossover points is swapped between the parents. If N is greater than or equal to three, then genetic information between at least two segments is exchanged. For instance, three segments are exchanged between parents if N is equal to five or six. The length of each segment might be different from each other and is determined by the positions of the crossover points.

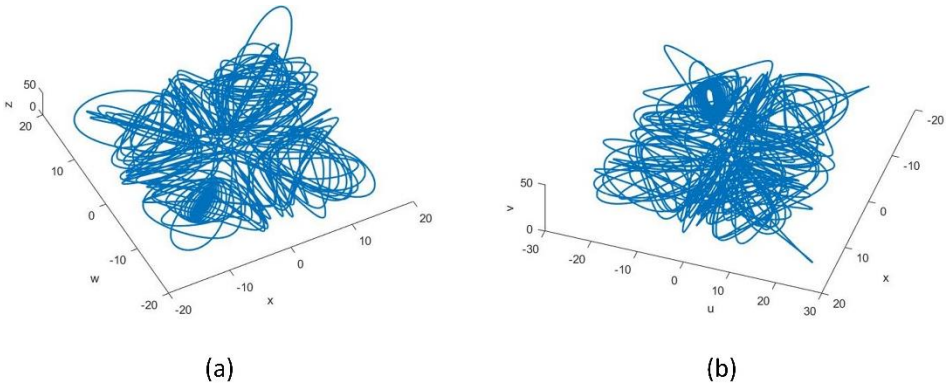

Figure 1. The Four-Wing Attractors of the 6D Hyperchaotic System

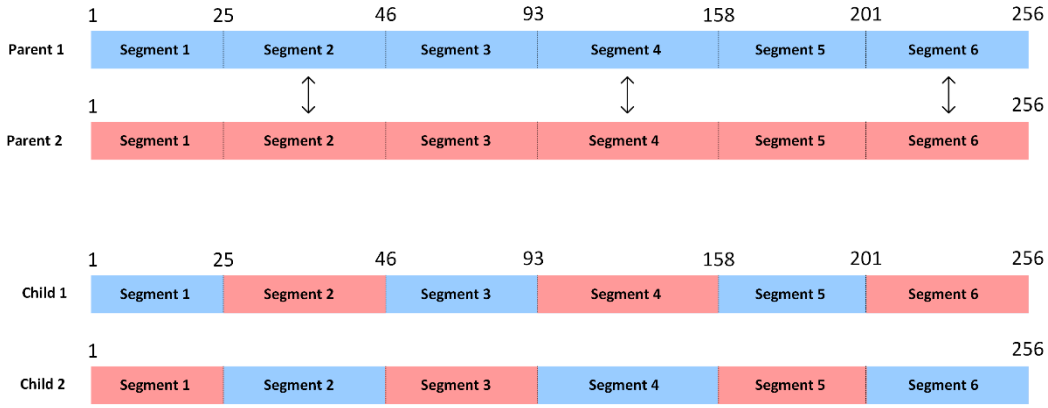

Figure 2. An Example of a 5-Point Crossover Operation

In the confusion stage of the proposed algorithm, two rounds of N-point crossover operation are implemented to shuffle the pixels of the plain image. First of all, parents should be selected from the rows and columns of the plain image to perform the crossover operation. The pseudo-random sequences generated by the state variables x and y in Eq. (1) are used to generate the row pairs and column pairs, respectively. The whole row or column can be regarded as chromosomes or parents and each pixel can be regarded as genes. The number of crossover points that specify the number of exchanged segments varies for different row and column pairs. The pseudo-random sequences produced with the help of the state variables z and w of Eq. (1) are utilized to calculate the number of crossover points of the row pairs and column pairs, respectively. After the determination of the pairs and the number of crossover points, the length and content of segments for each pair are identified. The position of the crossover points determines the length and content of segments on each chromosome. The state variables u and v in Eq. (1) create two pseudo-random sequences that determine the crossover positions for row and column pairs, respectively. In the second round, all of the initial conditions of the hyperchaotic map are varied and the above-mentioned processes are re-implemented. To sum up, the pixels are shuffled with the help of the N-point crossover operation whose shuffling parameters are calculated from the random sequences generated by Eq. (1). In Fig. 2, an illustration of a 5-point crossover operation example is shown. Parent 1 and Parent 2 could be rows or columns of the plain image with a length of 256 pixels. Randomly chosen five crossover points are 25, 46, 93, 158, and 201 which creates six segments. Segment 2, segment 4, and segment 6 are exchanged between Parent 1 and Parent 2 so that Child 1 and Child 2 are obtained as the new generation.

Key Generation for the Diffusion Stage

Hash functions are used to generate fixed-size output data from input of arbitrary size. As a member of the Secure Hash Algorithm (SHA) family, SHA-256 gives an output with a digest length of 256 bits for any given input. In this work, the SHA-256 hash value of the plain image is utilized in the key generation for the diffusion stage. The 256-bit hash value is divided into 16 blocks with a length of 16 bits as shown in Eq. (2).

$$K_i = \{K_1, K_2, \ldots, K_{16}\} \tag{2}$$

where i denotes the block number and each K consists of 16 bits. Mutation and crossover operations of genetic algorithms are used to further complicate the encryption key of the diffusion stage. Initially, the ith block's ith bit where i = 1,2,…,16 is flipped for mutation operation. Subsequently, a two-point crossover operation is implemented on the mutated blocks. The ith block is paired with the jth block where j=17–i and i = 1,2,…,8. Starting from the (i+1)th position, j-5 bits are swapped between the paired blocks. For example, K_1 and K_{16} blocks are paired with each other and eleven bits from the second position to the twelfth position are exchanged between the blocks. $K_{i'}$ is the generated 256-bit sequence which consists of 16 blocks after the genetic operations.

THE PROPOSED IMAGE ENCRYPTION ALGORITHM

The proposed image encryption algorithm consists of two main stages. In the first stage, the plain image's pixels are confused using the N-point crossover operation after its parameters are computed with the aid of a 6D hyperchaotic system. Additionally, the shuffled image is diffused with the keys which are obtained by applying mutation and crossover operations to the hash value of the plain image. The plain image P with a size of M x N is given as the input, where M and N are the numbers of rows and columns, respectively. The proposed encryption algorithm's steps are listed as follows:

Step 1. The 6D hyperchaotic system in Eq. (1) is iterated for 1000 + L times with the initial conditions x_0, y_0, z_0, w_0, u_0, v_0. The system parameters a, b, c, and the scaling factors σ_1 and σ_2 are selected properly so that the system is ensured to be hyperchaotic. A total of six pseudo-random sequences are produced by the state variables x, y, z, w, u, and v. The first 1000 values in those sequences are discarded to get rid of the transient effects.

Step 2. The rows and columns of the plain image are paired with each other with the help of the pseudo-random sequences generated by the state variables x and y, respectively. The elements in the obtained sequences $S_x = \{x_{1001}, x_{1002}, \dots, x_{1000+M}\}$ and $S_y = \{y_{1001}, y_{1002}, \dots, y_{1000+N}\}$ are sorted in ascending order. The positions of the sorted values are found in the sequences S_x and S_y so that the position arrays $R = (r_1, r_2, \dots, r_M)$ and $C = (c_1, c_2, \dots, c_N)$ can be obtained. Starting from the first element, every two adjacent elements in the position arrays R and C are grouped sequentially. Each group consisting of two rows or columns are the paired parents whose genes will be exchanged between them.

Step 3. The number of crossover points for each row and column pair is calculated using the pseudo-random sequences produced through the state variables z and w, respectively. A total of $M/2$ and $N/2$ crossover points are determined for the row and column pairs, respectively. The following equations shown in Eq. (3) and Eq. (4) are used to find the number of crossover points.

$$N_r = mod\left(\left\lfloor abs\big(z(i)\big) \times 10^{10} \right\rfloor, N_{max} - N_{min} + 1\right) + N_{min} \tag{3}$$

$$N_c = mod\left(\left\lfloor abs\big(w(j)\big) \times 10^{10} \right\rfloor, N_{max} - N_{min} + 1\right) + N_{min} \tag{4}$$

where N_r and N_c are the sequences which include the number of crossover points for row and column pairs, $i = 1001, 1002, \dots, 1000 + M/2$, and $j = 1001, 1002, \dots, 1000 + N/2$. $N_{min} \in [1,4]$ is the minimum N-point, $N_{max} \in [5,8]$ is the maximum N-point. N_{min} and N_{max} are calculated using the following formulas with the help of the 256-bit hash value of the plain image K_i.

$$N_{min} = mod(\textstyle\sum_{i=1}^{8} dec(K_i)\ , 4) + 1 \tag{5}$$

$$N_{max} = mod(\textstyle\sum_{i=9}^{16} dec(K_i)\ , 4) + 5 \tag{6}$$

where dec converts 16-bit binary numbers to decimal numbers.

Step 4. The positions of the crossover points are determined by the sequences generated using the state variables u and v, respectively. The positions of crossover points are selected between 2 and $M - 1$, and 2 and $N - 1$ for row and column pairs, respectively. The expressions for the crossover positions are given as the following equations.

$$P_r = mod\left(\left\lfloor abs\big(u(k)\big) \times 10^{10} \right\rfloor, M - 2\right) + 2 \tag{7}$$

$$P_c = mod\left(\left\lfloor abs\left(v(l)\right) \times 10^{10} \right\rfloor, N-2\right) + 2 \quad (8)$$

where P_r and P_c are the sequences which contain the positions of the crossover points, $k = 1001, 1002, \ldots, 1000 + \sum_i N_r(i)$ and $l = 1001, 1002, \ldots, 1000 + \sum_j N_c(j)$. Therefore, the maximum number of iterations L could be equal to the greatest of $(M/2)N_{max}$ or $(N/2)N_{max}$.

Step 5. The N-point crossover operation is implemented on the input image. The rows of the plain image are transformed using the position arrays R, the number of crossover points N_r and the positions of the crossover points P_r as described in Sect. 2.2. After transforming the rows, the columns of the plain image are modified in the same way using C, N_c and P_c.

Step 6. In the second round of the shuffling phase, the initial conditions x_0, y_0, z_0, w_0, u_0, v_0 are updated and the processes from Step 1 to Step 5 are re-implemented.

Step 7. The keys for the diffusion phase are obtained as discussed in Sect. 2.3. There are 16 keys with a length of 16 bits each. The following formulas are used to diffuse the shuffled image P_s with the generated keys $K_{i'}$.

$$C(1) = P_s(1) \oplus K_1(1{:}8) \quad (9)$$

$$C(2) = P_s(2) \oplus K_1(9{:}16) \quad (10)$$

where $K_1(1{:}8)$ represents the first eight bits of Key 1, $K_1(9{:}16)$ represents the last eight bits of Key 1. $P_s(1)$ and $P_s(2)$ are the first two pixels of the shuffled image. $C(1)$ and $C(2)$ are the first two pixels of the encrypted image. The equations in (11) and (12) are used to cipher the rest of the pixels of the shuffled image.

$$C(i) = K_k(1{:}8) \oplus P_s(i) \oplus C(i-2) \quad (11)$$

$$C(i+1) = K_k(9{:}16) \oplus P_s(i+1) \oplus C(i-1) \quad (12)$$

where C is the encrypted image, $k = mod\left(\frac{i+1}{2}, 16\right) + 1$ and $i = 3,5,7,9, \ldots, MN-1$. The proposed image encryption architecture is illustrated in Fig. 3. The decryption of the ciphered image can be accomplished by following the steps from 7 to 1 sequentially.

EXPERIMENTAL RESULTS AND ANALYZES

Lena, Cameraman, Mandril, and Peppers which are grayscale standard test images with sizes of 256x256 are used in the experimental study. The following analyzes are performed for the test images: Keyspace and key sensitivity, histogram, correlation (adjacent pixel), information entropy, differential, data loss, noise attack, and computational time. The results of these analyzes are compared with the results of many other recently published studies.

Keyspace and Key Sensitivity Analysis

The key space can be found by calculating the total number of all possible keys which are used in the encryption process. To resist an exhaustive search attack that tries every possible key, the key space must be larger than 2^{100} (Alvarez and Li, 2006). The initial values of the 6D hyperchaotic system in the first and second rounds of the shuffling are determined as the secret keys. The initial values are real numbers with a precision of 10^{-15}. The key space can be computed as in Eq. (13) for a plain image with a size of 256×256 pixels.

$$Key\ Space = (10^{15})^6 \times (10^{15})^6 \approx 2^{597} \gg 2^{100} \quad (13)$$

Table 1 shows the comparison of key space for different studies. As seen in the table, this study's key space is larger than the referred works' key spaces. The proposed algorithm's key space is sufficiently large to resist an exhaustive-search attack. A reliable image encryption system must be sensitively dependent on the secret keys, which means that a tiny deflection in any key results in a completely unrecognizable decrypted image. The initial conditions of the 6D system which are part of the secret keys are chosen as follows: $x_0 = 0$, $y_0 = 1$, $z_0 = 20$, $w_0 = 0.2$, $u_0 = 0.2$, $v_0 = 0.2$ for the first round of shuffling and $x_{02} = 0.1$, $y_{02} = 2$, $z_{02} = 21$, $w_{02} = 0.21$, $u_{02} = 0.21$, $v_{02} = 0.21$ for the second round of the shuffling. Figure 4 shows the plain images of Lena, Cameraman, Mandril, and Peppers, their

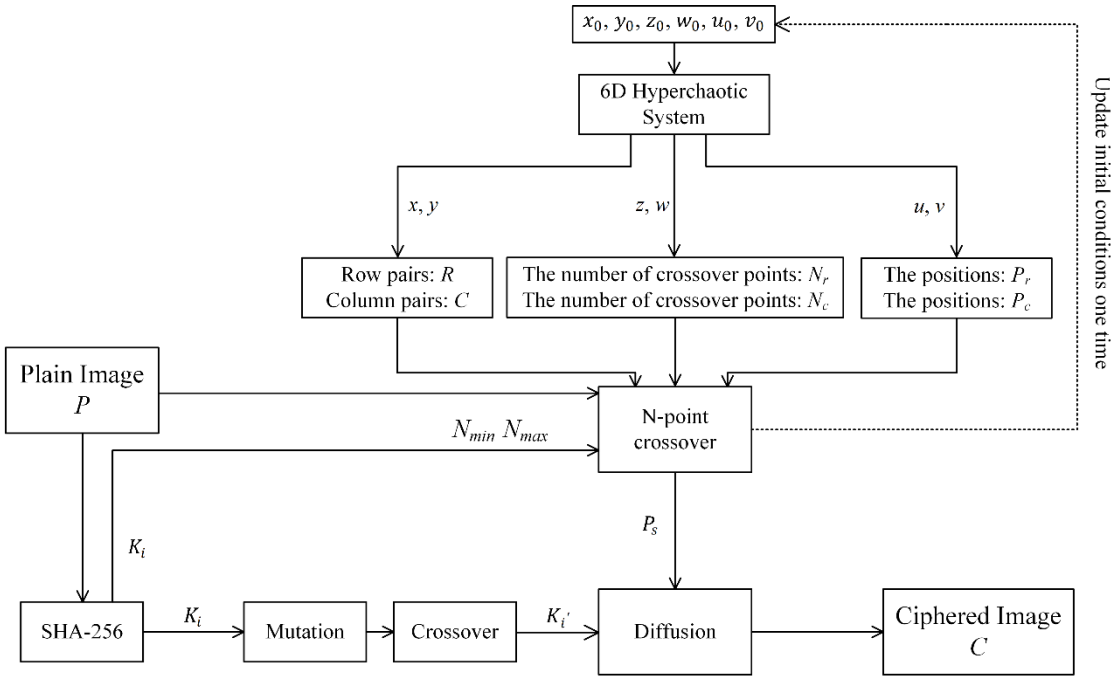

Figure 3. The Proposed Image Encryption Architecture

Table 1. Comparison of Keyspace

	This study	(Q. Zhang and Han, 2021)	(Zhou and Wang, 2020)	(Hui et al., 2021)	(Zeng and Wang, 2021)	(Chai et al., 2021)
Keyspace	2^{597}	2^{536}	2^{399}	$10^{60} \approx 2^{199}$	2^{512}	2^{280}

corresponding encrypted images, and their corresponding decrypted images with the correct keys. To illustrate the encryption method's sensitivity to the secret keys, encrypted test images are decrypted with slightly altered keys. As shown in Fig. 5, only one secret key is changed very slightly while other secret keys are kept the same for each test image. These decrypted images are incorrectly deciphered and entirely different from the ones which are shown in Fig. 4. Therefore, it can be said that this image encryption scheme is sensitive to the selected secret keys.

Histogram Analysis

The values of pixels in the encrypted images should be uniformly distributed to resist statistical attacks which aim to obtain information by analyzing the distribution of the pixels. The histogram graphs of the test images for plain and encrypted versions are illustrated in Fig. 6. The suggested encryption method provides uniform distribution for the encrypted images so that statistical attacks can be made infeasible.

To quantify how uniformly the pixels are distributed for the plain and encrypted test images, the variance of histograms can be calculated. The variance value should be smaller for a securely encrypted image as an indication of uniform pixel distribution. Ideally, the variance is zero for an 8-bit grayscale image if all 256 tonal values have the same repetition frequency. Table 2 lists the variance values of the histogram plots for plain and encrypted test images. Table 2 indicates that the variances are significantly reduced after the encryption process.

Table 2. Histogram Variances for Plain and Encrypted Images

Test image	Plain	Encrypted
Lena	30785.960	234.047
Cameraman	111408.494	247.890
Mandril	59087.020	253.663
Peppers	31763.255	250.776

Correlation Analysis

Since a plain image is a visualization of meaningful data, neighboring pixels of the plain image are highly similar in horizontal, vertical, and diagonal directions. Thus, the plain image's adjacent pixels are highly correlated. One of the

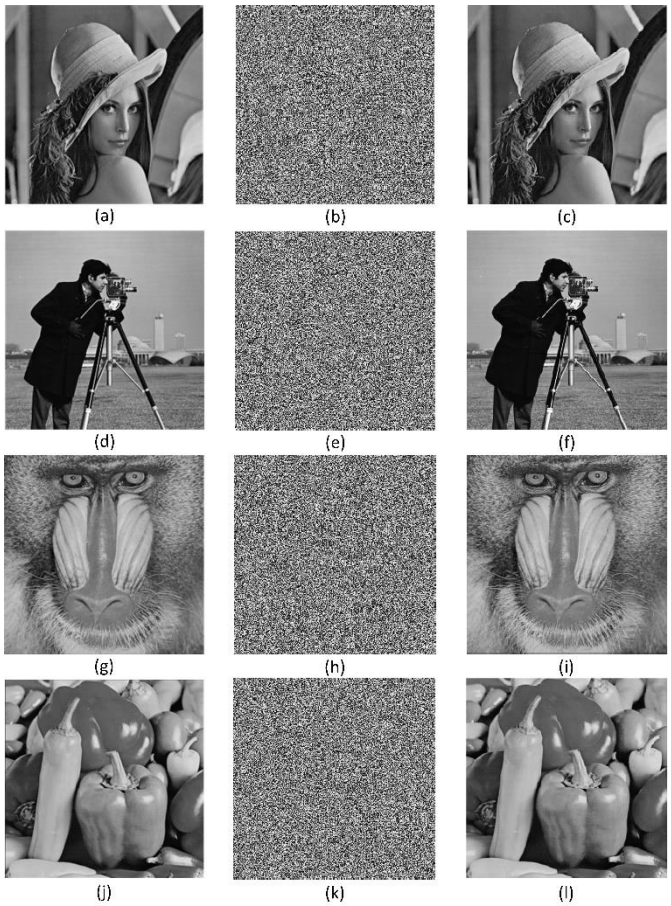

Figure 4. Encryption and Decryption Results with the Correct Keys

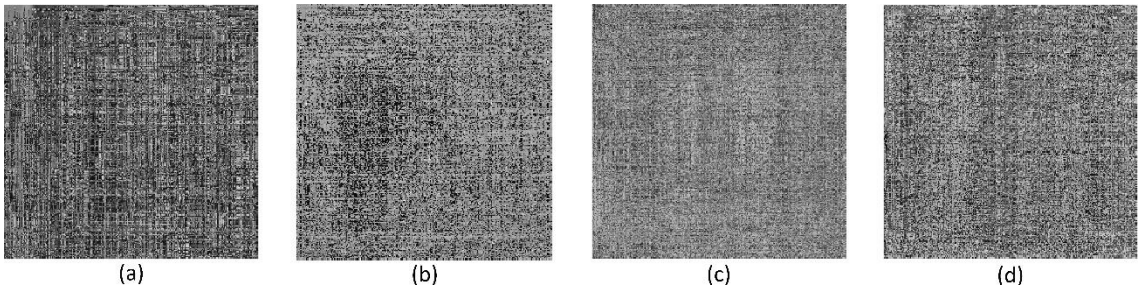

Figure 5. Decryption with Incorrect Keys **a.** Decrypted Lena with $y_0 = 1 + 10^{-15}$ **b.** Decrypted Cameraman with $v_{02} = 0.21 - 10^{-15}$ **c.** Decrypted Mandril with $w_0 = 0.2 + 10^{-15}$ **d.** Decrypted Peppers with $x_{02} = 0.1 - 10^{-15}$

the aims of the image encryption methods is to reduce this correlation between adjacent pixels. A total of 2500 adjacent pixel pairs are selected randomly from the plain and encrypted test images in horizontal, vertical, and diagonal directions. A high correlation between neighboring pixels in all directions can be identified from Fig. 7(a), 7(c), 7(e), and 7(g) for the plain images. As is observed in Fig. 7(b), 7(d), 7(f), and 7(h), however, the strong correlation is significantly reduced in all directions after the encryption process. Thus, it is safe to say that this encryption method can resist statistical attacks.

The correlation coefficient can be used as an evaluation metric to determine the correlation between adjacent pixels and it can be expressed as in Eq. (14).

$$\rho = \frac{\sum_{i=1}^{n}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{n}(x_i - E(x))^2 \sum_{i=1}^{n}(y_i - E(y))^2}} \tag{14}$$

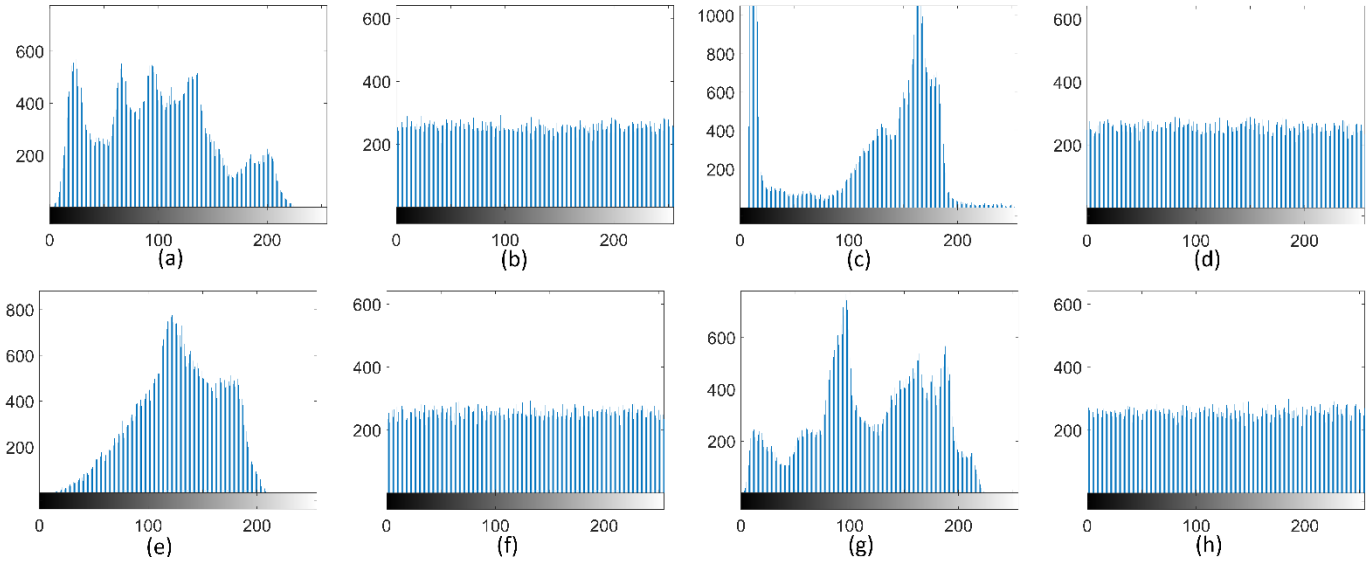

Figure 6. Histogram Graphs **a.** Lena **b.** Encrypted Lena **c.** Cameraman **d.** Encrypted Cameraman **e.** Mandril **f.** Encrypted Mandril **g.** Peppers **h.** Encrypted Peppers

Table 3. Average Correlation Coefficients for Test Images and Comparison with Other Works

Test image	Direction	Plain	Encrypted	(Patro et al., 2019)	(Niu et al., 2020)	(Gupta et al., 2021b)	(J. Wang, Zhi, Chai, and Lu, 2021)	(Xingyuan Wang, Lin, and Li, 2021)
Lena	Horizontal	0.940712	-0.008011	- 0.0089	0.0055	0.0057	0.0084	**0.0006**
	Vertical	0.969795	0.004080	0.0016	0.0305	0.0087	−0.0039	**0.0003**
	Diagonal	0.917030	**0.001165**	0.0068	−0.0043	0.0387	−0.0013	0.0020
Cameraman	Horizontal	0.933703	**0.000178**	- 0.0013	-	-	-	0.0007
	Vertical	0.960281	-0.004402	**0.0017**	-	-	-	0.0024
	Diagonal	0.909464	-0.001455	0.0070	-	-	-	**0.0003**
Mandril	Horizontal	0.890513	**0.003920**	-	−0.0138	0.0062	-	-
	Vertical	0.858449	**0.007558**	-	−0.0267	0.0090	-	-
	Diagonal	0.816526	**-0.001011**	-	−0.0072	0.0851	-	-
Peppers	Horizontal	0.962784	**-0.002207**	-	0.0190	-	-	0.0026
	Vertical	0.970404	0.003186	-	0.0029	-	-	**0.0010**
	Diagonal	0.937218	**-0.000596**	-	0.0334	-	-	0.0029

where $E(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$, $E(y) = \frac{1}{n}\sum_{i=1}^{n} y_i$, x and y denote the values of the adjacent pixels and n represents the number of randomly chosen pixels. The correlation coefficient ρ must approximate 1 and 0 for plain and encrypted images, respectively. The correlation coefficients in horizontal, vertical, and diagonal directions for plain and encrypted test images can be calculated using Eq. (14). 2500 different adjacent pixel pairs are randomly selected in all directions to compute ρ. This selection and computation procedure is repeated 10 times. The average values of the calculated correlation coefficients are listed for the plain and encrypted test images in Table 3. It can be noticed from Table 3 that the plain images have very high correlation values which are close to 1 in all directions. The correlation coefficient values for the encrypted images, however, are close to zero in all directions. A significant reduction in the correlation coefficients results as a consequence of the applied image encryption method. In Table 3, the suggested method is also compared with some recent studies (Gupta et al., 2021b; Niu et al., 2020; Patro et al., 2019; J. Wang et al., 2021; X. Wang et al., 2021) in terms of correlation coefficients of the encrypted images of the same size. The best correlation coefficient values are marked in bold for the encrypted images in the table. The presented image encryption method provides the best correlation coefficient reduction performance for seven out of twelve calculations.

Information Entropy Analysis

Information entropy can be used as an evaluation metric to measure the randomness of the plain and encrypted images. The information entropy value can be ideally 8 for an 8-bit encrypted grayscale image, whereas its value is smaller for plain images. The information entropy of a signal s can be calculated as expressed in Eq. (15).

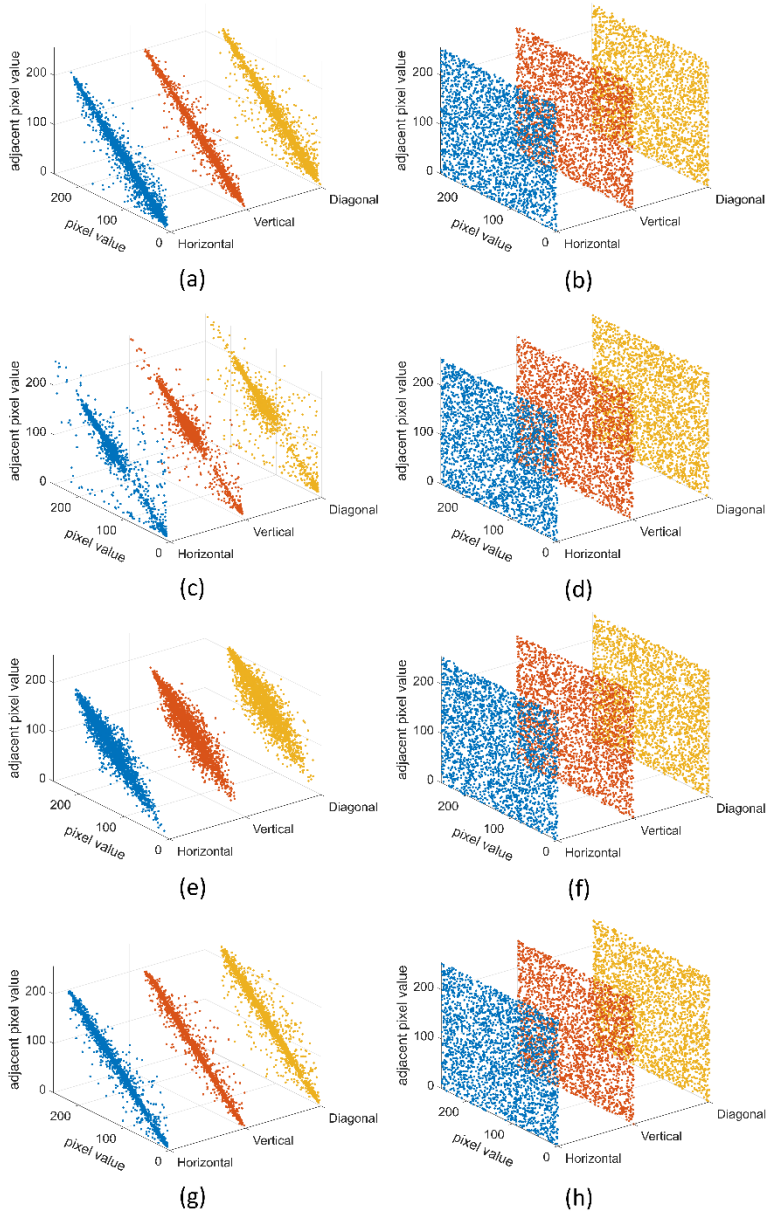

Figure 7. Correlation Between Adjacent Pixels in Horizontal, Vertical, Diagonal directions **a.** Lena **b.** Encrypted Lena **c.** Cameraman **d.** Encrypted Cameraman **e.** Mandril **f.** Encrypted Mandril **g.** Peppers **h.** Encrypted Peppers

Table 4. Information Entropies for Test Images and Comparison with Other Works

Test image	Plain	Encrypted	(Patro et al., 2019)	(Gupta et al., 2021b)	(J. Wang et al., 2021)	(X. Wang, Zhu, and Zhang, 2018)
Lena	7.5683	7.9974	7.9974	7.9971	7.9973	7.9971
Cameraman	7.0097	7.9973	7.9972	-	-	7.9971
Mandril	7.2283	7.9972	-	7.9969	-	-
Peppers	7.5798	7.9972	-	-	-	7.9968

$$IE = -\sum_{i=0}^{255} P(s_i)\, log_2\, P(s_i) \quad (15)$$

where $P(s_i)$ is the probability of frequency of the symbol s_i. To compute the information entropy of the plain and encrypted test images, $P(s_i)$ can be considered as the probability of occurrence of each pixel value. Table 4 lists the calculated information entropy values for plain and encrypted test images. The information entropy values for the encrypted images are larger than 7.997 which is quite close to the ideal value. This indicates the good randomness of the encrypted images. Compared with existing studies (Gupta et al., 2021b; J. Wang et al., 2021; Patro et al., 2019; X. Wang et al., 2018), this method presents better information entropy results.

Table 5. NPCR and UACI Values of Test Images for Five Different Pixel Changes

Test image	Position of pixel	Pixel value change	NPCR (%)	UACI (%)
Lena	(1,24)	151→150	99.64447	33.35464
	(79,13)	70→71	99.59259	33.50028
	(125,132)	100→99	99.58801	33.50730
	(199,201)	93→94	99.64447	33.24368
	(250,1)	29→28	99.58954	33.36937
Average			99.61182	33.39505
Cameraman	(1,1)	156→155	99.57886	33.57422
	(50,50)	174→175	99.60632	33.31693
	(100,120)	12→13	99.63074	33.45505
	(150,170)	162→163	99.61243	33.38579
	(256,256)	113→112	99.59717	33.45018
Average			99.60510	33.43643
Mandril	(10,250)	124→123	99.62616	33.50114
	(49,155)	130→131	99.59564	33.55044
	(125,3)	104→103	99.61395	33.40848
	(202,256)	135→136	99.59869	33.39330
	(245,23)	159→158	99.57428	33.37628
Average			99.60174	33.44593
Peppers	(25,1)	85→86	99.56360	33.27225
	(115,44)	43→42	99.62006	33.44840
	(177,177)	131→132	99.63684	33.53672
	(200,9)	68→69	99.61853	33.28563
	(250,187)	162→161	99.57733	33.39371
Average			99.60327	33.38734
Overall average			**99.60549**	**33.41619**

Differential Analysis

Differential attacks aim to detect a relationship between two ciphered images whose corresponding plain images own one pixel difference. Even if there is a slight difference between two plain images, a robust encryption method should generate completely different encrypted images for those plain images. To defend against differential attacks, sensitivity to the plain image should be provided. Two performance metrics are typically used for differential analysis: Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI). NPCR is used to calculate the ratio of the number of different pixels between two encrypted images. Similarly, UACI computes the percentage of the mean difference in intensities between two encrypted images. NPCR and UACI values can be found using the equations in Eq. (16) and Eq. (17) for an image with a size of M x N.

$$NPCR = \frac{1}{M \times N} \sum_{j=1}^{N} \sum_{i=1}^{M} D(i,j) \times 100\ \% \tag{16}$$

$$UACI = \frac{1}{M \times N \times 255} \sum_{j=1}^{N} \sum_{i=1}^{M} |E_1(i,j) - E_2(i,j)| \times 100\ \% \tag{17}$$

where E_1 and E_2 are two images that are encrypted with the proposed algorithm, whose corresponding images contain only one pixel difference. The elements of D, which is an M x N dimensional matrix, are given as follows.

$$D(i,j) = \begin{cases} 1 & if \quad E_1(i,j) \neq E_2(i,j) \\ 0 & if \quad E_1(i,j) = E_2(i,j) \end{cases} \tag{18}$$

NPCR and UACI values of the test images are calculated using the equations given in Eq. (16), Eq. (17) and Eq. (18) and are presented in Table 5. One pixel is arbitrarily selected from each test image and the pixel's value is increased or decreased by one. The plain image and the slightly modified image are both encrypted with the same secret keys to obtain E_1 and E_2. This process is repeated for 5 separate pixels of each test image. The average of five different NPCR values are listed as follows: 99.61182%, 99.60510%, 99.60174%, and 99.60327% for Lena, Cameraman, Mandril, and Peppers images, respectively. The average UACI values are computed as 33.39505%, 33.43643%, 33.44593%, and 33.38734% for Lena, Cameraman, Mandril, and Peppers images, respectively. Table 6 illustrates the comparison of average NPCR and UACI values with recently published studies (Karawia and Elmasry, 2021; X. Wang et al., 2021). The overall average of NPCR and UACI are calculated as 99.60549% and 33.41619%. The ideal

Table 6. Comparison of NPCR and UACI Values with Recent Studies

Test image	This study		(Karawia and Elmasry, 2021)		(X. Wang et al., 2021)	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Lena	99.61182	33.39505	99.6099	33.4025	99.6078	33.5309
Cameraman	99.60510	33.43643	99.6145	33.5832	99.6215	33.3309
Mandril	99.60174	33.44593	99.6140	33.4384	-	-
Peppers	99.60327	33.38734	99.6368	33.4185	99.6200	33.4506

Table 7. PSNR And MSE Values for the Test Images

Test image	Data loss ratio	MSE	PSNR (dB)
Lena	1/64	119.802	27.346
	1/8	1188.124	17.382
	1/4	2459.674	14.222
Cameraman	1/64	133.799	26.866
	1/8	1187.897	17.383
	1/4	2503.545	14.145
Mandril	1/64	108.735	27.767
	1/8	901.682	18.580
	1/4	1767.412	15.657
Peppers	1/64	115.340	27.511
	1/8	920.479	18.490
	1/4	1897.099	15.350

values suggested in the literature (Y. Li, Wang, and Chen, 2017) are 99.61% and 33.46% for NPCR and UACI, respectively. The overall average values are pretty close to the ideal ones which indicate that the proposed encryption scheme can effectively resist differential attacks.

Data Loss Analysis

The data of the encrypted image may be lost during the process of transmission as a consequence of attacks. The attacked and disrupted encrypted image should be decrypted as accurately as possible. Data loss analysis is performed by clipping a certain amount of the encrypted images deliberately. The clipped images are decrypted with the correct keys and the decrypted images are compared with the plain images. Peak signal-to-noise ratio (PSNR) can be used as a parameter to compare plain images and decrypted images. For an 8-bit grayscale image, PSNR can be calculated as in Eq. (19).

$$PSNR = 10\,log_{10}\frac{(2^8-1)^2}{MSE}\ (dB) \tag{19}$$

where *MSE* stands for mean squared error and it can be defined as in Eq. (20) for an image with a size of *M* x *N*.

$$MSE = \frac{1}{MN}\sum_{j=1}^{N}\sum_{i=1}^{M}\big(P(i,j) - D(i,j)\big)^2 \tag{20}$$

where *P* is the plain image and *D* is the decrypted image. Theoretically, if the plain image and the decrypted image are the same, then MSE becomes zero and PSNR goes to infinity. PSNR value will be high if the plain image and the decrypted image are very similar to each other. To test the effect of data loss, 1/64, 1/8, and 1/4 of the upper left corner of the encrypted image are clipped. The data loss analysis visual results are presented in Fig. 8 for Lena and Cameraman test images. The PSNR and MSE values for all of the test images are also calculated and presented in Table 7.

As shown in Fig. 8, Lena and Cameraman images are recovered from the clipped encrypted images. The recovered images are more recognizable for lower data loss ratios due to the lower mean squared error rates. Table 7 shows that if the data loss ratio increases, the PSNR value decreases. The highest PSNR values are obtained as 27.767 dB, 18.580 dB, and 15.657 dB for 1/64, 1/8, and 1/4 data losses, respectively. A comparison of PSNR is shown in Table 8, where the encrypted Lena image is cropped by 1/8 or 1/4 from the upper left corner. Comparison with the recently published studies (Karawia and Elmasry, 2021; Xingyuan Wang and Li, 2021; T. Wang and Wang, 2020) shows that the suggested method appears to perform better in terms of recovering images with data loss.

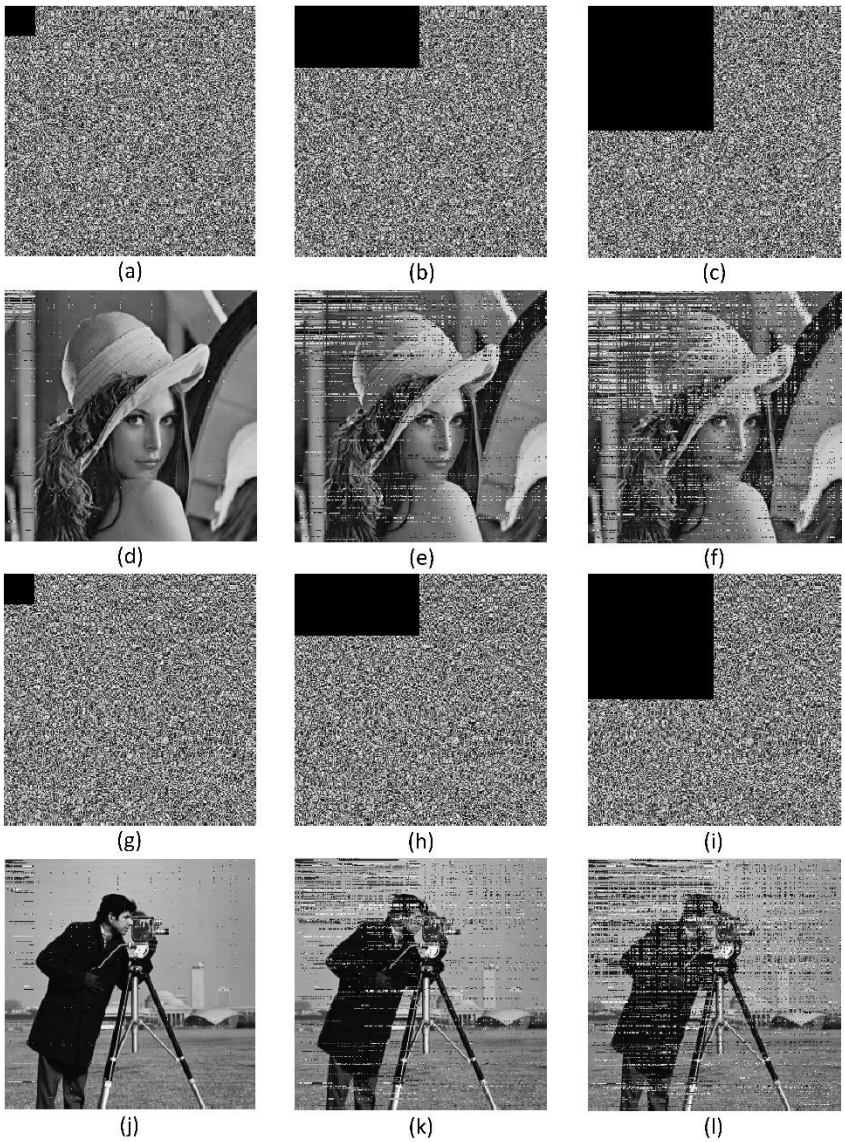

Figure 8. Data Loss Analysis **a.** Lena with 1/64 Loss **b.** Lena with 1/8 Loss **c.** Lena with 1/4 Loss **d.** Decrypted Lena with 1/64 Loss **e.** Decrypted Lena with 1/8 Loss **f.** Decrypted Lena with 1/4 Loss **g.** Cameraman with 1/64 Loss **h.** Cameraman with 1/8 Loss **i.** Cameraman with 1/4 Loss **j.** Decrypted Cameraman with 1/64 Loss **k.** Decrypted Cameraman with 1/8 Loss **l.** Decrypted Cameraman with 1/4 Loss

Table 8. Comparison of PSNR (dB) Value of Lena image with Recent Studies

Test image	Data loss ratio	This study	(T. Wang and Wang, 2020)	(Karawia and Elmasry, 2021)	(Xingyuan Wang and Li, 2021)
Lena	1/8 at the upper left corner	17.382	8.6402	8.8	13.6449
	1/4 at the upper left corner	14.222	8.0511	8.1	10.6747

Noise Attack Analysis

The noise in the transmitted image data might be a problem. A robust image encryption system should recover the noisy encrypted image to a large extent. Possible noise attacks can be simulated by intentionally adding various noise types with different intensities. The encrypted test images are contaminated with salt and pepper noise (SPN) with an intensity of 0.05, 0.1, and 0.15; speckle noise (SN) with an intensity of 0.001, 0.01, and 0.02; and Poisson noise (PN) to test the suggested encryption method's resistance against noise attacks. The noise attack simulation results of Mandril and Peppers test images are presented in Fig. 9 for various types of noises with different intensities. The most identifiable image is detected in Fig. 9(d), whereas the least recognizable one is given in Fig. 9(l). The PSNR

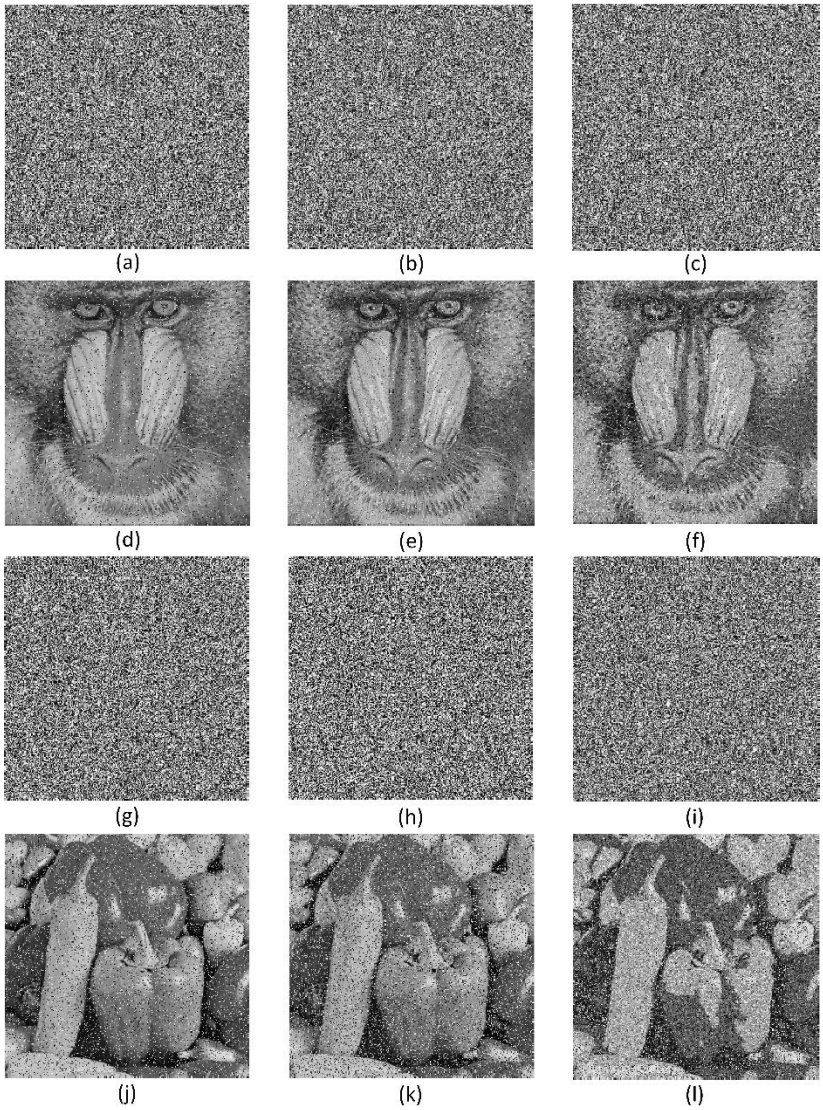

Figure 9. Noise Attack Analysis **a.** Mandril 0.05 SPN **b.** Mandril 0.001 SN **c.** Mandril PN **d.** Decrypted Mandril 0.05 SPN **e.** Decrypted Mandril 0.001 SN **f.** Decrypted Mandril PN **g.** Peppers 0.10 SPN **h.** Peppers 0.15 SPN **i.** Peppers 0.02 SN **j.** Decrypted Peppers 0.10 SPN **k.** Decrypted Peppers 0.02 SN **l.** Decrypted Peppers 0.02 SN

Table 9. PSNR (dB) Values for Various Noise Attacks

Test image	Noise type and intensity						
	SPN 0.05	**SPN 0.1**	**SPN 0.15**	**SN 0.001**	**SN 0.01**	**SN 0.02**	**PN**
Lena	18.660	15.824	14.088	18.839	14.347	13.209	15.116
Cameraman	18.589	15.528	13.872	18.448	14.080	12.959	14.952
Mandril	19.752	17.020	15.387	19.640	15.504	14.392	16.347
Peppers	19.047	16.124	14.415	19.502	15.067	13.834	15.973

Table 10. Comparison of PSNR (dB) Value of Lena Images for Different Noise Attacks

Test image	Noise type and intensity	This study	(T. Wang and Wang, 2020)	(J. Wang et al., 2021)	(Xingyuan Wang and Chen, 2021)
Lena (256x256)	SN 0.00000143	62.999	-	41.4243	-
	SPN 0.00003	55.288	-	52.3022	-
	SPN 0.10	15.824	8.6674	-	-
	SPN 0.15	14.088	8.4304	-	-
Lena (512x512)	SPN 0.05	18.096	-	-	17.101
	SPN 0.10	15.253	-	-	14.286
	SN 0.02	12.471	-	-	10.357
	PN	14.390	-	-	10.827

values of these images are 19.752 dB and 13.834 dB, respectively. Table 9 lists the PSNR values which are calculated between plain images and the noisy encrypted images. The cryptosystem's resistance varies as the noise type changes but the PSNR value always decreases as the noise intensity increases. Also, the noise performance of the suggested method is compared with some recent studies (J. Wang et al., 2021; Xingyuan Wang and Chen, 2021; T. Wang and Wang, 2020) in terms of PSNR values in Table 10. The same size of encrypted Lena images is contaminated with the same types and intensities of noise. This study's noise performance is better than that of (J. Wang et al., 2021) for both SN and SPN. Also, this study's resistance against SPN with intensities of 0.10 and 0.15 is more robust than that of (T. Wang and Wang, 2020). Moreover, the suggested method outperforms the method given in (Xingyuan Wang and Chen, 2021) for SPN, SN, and PN attacks to a 512x512 Lena image.

Computational Time Analysis

The proposed image encryption algorithm is executed on MATLAB 2017b using a PC with an Intel Core 2.80 GHz processor and 16 GB RAM. Average encryption and decryption times are found to be 1.429 seconds and 1.148 seconds for a grayscale image of size 256x256, respectively.

CONCLUSIONS

In this work, a new image encryption method, which is based on a 6D hyperchaotic map and genetic operators, is presented. The 6D hyperchaotic map's state variables are utilized to generate the parameters of the N-point crossover operation. The pseudo-random sequences generated by those state variables are used to pair rows and columns with each other and to determine the number of crossover points and positions. The maximum and minimum values of N are calculated using the SHA-256 hash value of the original image. The 256-bit hash value is also modified by mutation and two-point crossover operations for being used in the diffusion stage. The initial values of the 6D hyperchaotic, which increase the key space, are used as the secret keys. High key space and the key sensitivity to the secret keys protect against exhaustive search attacks. The test images' uniformly distributed histograms show that statistical attacks are infeasible. Also, the correlation between the adjacent pixels of the plain images in horizontal, vertical, and diagonal directions is significantly reduced. For each test image, an information entropy value greater than 7.997 is obtained. This is an indication of the good randomness of the encrypted images. Differential analysis proves that the proposed method's overall NPCR and UACI values are pretty close to the ideal values. Thus, the proposed encryption algorithm can resist differential attacks effectively. Finally, data loss and noise attack analyses prove that the proposed method can effectively recover the encrypted images with data loss or noise. The analysis results prove that the suggested encryption technique can be used for the safe transmission of images. The analysis results of the proposed scheme are also compared with several recently published state-of-the-art works. In most of the comparisons, this method outperforms the others. In future work, the main idea behind the suggested method will be improved and applied to color images.

REFERENCES

Alvarez, G., & Li, S. (2006). Some Basic Cryptographic Requirements for Chaos-based Cryptosystems. *International Journal of Bifurcation and Chaos, 16*(08), 2129-2151. doi:10.1142/s0218127406015970

Boriga, R., Dăscălescu, A. C., & Priescu, I. (2014). A new hyperchaotic map and its application in an image encryption scheme. *Signal Processing: Image Communication, 29*(8), 887-901. doi:https://doi.org/10.1016/j.image.2014.04.001

Cao, C., Sun, K., & Liu, W. (2018). A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Processing, 143*, 122-133. doi:https://doi.org/10.1016/j.sigpro.2017.08.020

Chai, X., Zhi, X., Gan, Z., Zhang, Y., Chen, Y., & Fu, J. (2021). Combining improved genetic algorithm and matrix semi-tensor product (STP) in color image encryption. *Signal Processing, 183*, 108041. doi:https://doi.org/10.1016/j.sigpro.2021.108041

Chen, X., Qian, S., Yu, F., Zhang, Z., Shen, H., Huang, Y., Du, S. (2020). Pseudorandom Number Generator Based on Three Kinds of Four-Wing Memristive Hyperchaotic System and Its Application in Image Encryption. *Complexity, 2020*, 8274685. doi:10.1155/2020/8274685

Cheng, G., Wang, C., Chen, H. J. I. J. o. B., & Chaos. (2019). A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture. *International Journal of Bifurcation and Chaos, 29*(09), 1950115. doi:https://doi.org/10.1142/S0218127419501153

Demirtaş, M. (2022). A new RGB color image encryption scheme based on cross-channel pixel and bit scrambling using chaos. *Optik, 265*, 169430. doi:https://doi.org/10.1016/j.ijleo.2022.169430

Dong, E., Yuan, M., Du, S., & Chen, Z. (2019). A new class of Hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator. *Applied Mathematical Modelling, 73*, 40-71. doi:https://doi.org/10.1016/j.apm.2019.03.037

Gao, X., Yu, J., Banerjee, S., Yan, H., & Mou, J. (2021). A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion. *Scientific Reports, 11*(1), 15737. doi:10.1038/s41598-021-94748-7

Grassi, G., Severance, F. L., & Miller, D. A. (2009). Multi-wing hyperchaotic attractors from coupled Lorenz systems. *Chaos, Solitons & Fractals, 41*(1), 284-291. doi:https://doi.org/10.1016/j.chaos.2007.12.003

Guesmi, R., Ben Farah, M. A., Kachouri, A., & Samet, M. (2016). Hash key-based image encryption using crossover operator and chaos. *Multimedia Tools and Applications, 75*(8), 4753-4769. doi:10.1007/s11042-015-2501-0

Gupta, M., Gupta, K. K., & Shukla, P. K. (2021a). Session key based fast, secure and lightweight image encryption algorithm. *Multimedia Tools and Applications, 80*(7), 10391-10416. doi:10.1007/s11042-020-10116-z

Gupta, M., Gupta, K. K., & Shukla, P. K. (2021b). Session key based novel lightweight image encryption algorithm using a hybrid of Chebyshev chaotic map and crossover. *Multimedia Tools and Applications, 80*(25), 33843-33863. doi:10.1007/s11042-021-11160-z

Hui, Y., Liu, H., & Fang, P. (2021). A DNA image encryption based on a new hyperchaotic system. *Multimedia Tools and Applications*. doi:10.1007/s11042-021-10526-7

Jia, Q. (2007). Hyperchaos generated from the Lorenz chaotic system and its control. *Physics Letters A, 366*(3), 217-222. doi:https://doi.org/10.1016/j.physleta.2007.02.024

Karawia, A. A., & Elmasry, Y. A. (2021). New Encryption Algorithm Using Bit-Level Permutation and Non-Invertible Chaotic Map. *IEEE Access, 9*, 101357-101368. doi:10.1109/ACCESS.2021.3096995

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091-8126. doi:10.1007/s11042-020-10139-6

Kaur, M., & Kumar, V. (2020). A Comprehensive Review on Image Encryption Techniques. *Archives of Computational Methods in Engineering, 27*(1), 15-43. doi:10.1007/s11831-018-9298-8

Kaur, M., Singh, D., & Kumar, V. (2020). Color image encryption using minimax differential evolution-based 7D hyper-chaotic map. *Applied Physics B, 126*(9), 147. doi:10.1007/s00340-020-07480-x

Li, P., Xu, J., Mou, J., & Yang, F. (2019). Fractional-order 4D hyperchaotic memristive system and application in color image encryption. *EURASIP Journal on Image and Video Processing, 2019*(1), 22. doi:10.1186/s13640-018-0402-7

Li, Y., Wang, C., & Chen, H. (2017). A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Optics and Lasers in Engineering, 90*, 238-246. doi:https://doi.org/10.1016/j.optlaseng.2016.10.020

Luo, Y., Zhou, R., Liu, J., Cao, Y., & Ding, X. (2018). A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map. *Nonlinear Dynamics, 93*(3), 1165-1181. doi:10.1007/s11071-018-4251-9

Mohamed, H. G., ElKamchouchi, D. H., & Moussa, K. H. (2020). A Novel Color Image Encryption Algorithm Based on Hyperchaotic Maps and Mitochondrial DNA Sequences. *Entropy, 22*(2), 158.

Mozaffari, S. (2018). Parallel image encryption with bitplane decomposition and genetic algorithm. *Multimedia Tools and Applications, 77*(19), 25799-25819. doi:10.1007/s11042-018-5817-8

Muthu, J. S., & Murali, P. (2021). Review of Chaos Detection Techniques Performed on Chaotic Maps and Systems in Image Encryption. *SN Computer Science, 2*(5), 392. doi:10.1007/s42979-021-00778-3

Natiq, H., Al-Saidi, N. M. G., Said, M. R. M., & Kilicman, A. (2018). A new hyperchaotic map and its application for image encryption. *The European Physical Journal Plus, 133*(1), 6. doi:10.1140/epjp/i2018-11834-2

Niu, Y., Zhou, Z., & Zhang, X. (2020). An image encryption approach based on chaotic maps and genetic operations. *Multimedia Tools and Applications, 79*(35), 25613-25633. doi:10.1007/s11042-020-09237-2

Patro, K. A. K., Acharya, B., & Nath, V. (2019). Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps. *Microsystem Technologies, 25*(12), 4593-4607. doi:10.1007/s00542-019-04395-2

Sun, S., Guo, Y., & Wu, R. (2019). A Novel Image Encryption Scheme Based on 7D Hyperchaotic System and Row-column Simultaneous Swapping. *IEEE Access, 7*, 28539-28547. doi:10.1109/ACCESS.2019.2901870

Wang Fa-Qiang, L. C.-X. (2006). Hyperchaos evolved from the Liu chaotic system. *Chinese Physics, 15*(5), 963-968. doi:10.1088/1009-1963/15/5/016

Wang, J., Zhi, X., Chai, X., & Lu, Y. (2021). Chaos-based image encryption strategy based on random number embedding and DNA-level self-adaptive permutation and diffusion. *Multimedia Tools and Applications, 80*(10), 16087-16122. doi:10.1007/s11042-020-10413-7

Wang, T., & Wang, M.-h. (2020). Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Optics & Laser Technology, 132*, 106355. doi:https://doi.org/10.1016/j.optlastec.2020.106355

Wang, X., & Chen, X. (2021). An image encryption algorithm based on dynamic row scrambling and Zigzag transformation. *Chaos, Solitons & Fractals, 147*, 110962. doi:https://doi.org/10.1016/j.chaos.2021.110962

Wang, X., & Li, Y. (2021). Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and DNA sequence. *Optics and Lasers in Engineering, 137*, 106393. doi:https://doi.org/10.1016/j.optlaseng.2020.106393

Wang, X., Lin, S., & Li, Y. (2021). Bit-level image encryption algorithm based on BP neural network and gray code. *Multimedia Tools and Applications, 80*(8), 11655-11670. doi:10.1007/s11042-020-10202-2

Wang, X., & Xu, D. (2014). Image encryption using genetic operators and intertwining logistic map. *Nonlinear Dynamics, 78*(4), 2975-2984. doi:10.1007/s11071-014-1639-z

Wang, X., Zhu, X., & Zhang, Y. (2018). An Image Encryption Algorithm Based on Josephus Traversing and Mixed Chaotic Map. *IEEE Access, 6*, 23733-23746. doi:10.1109/ACCESS.2018.2805847

Wu, J., Shi, J., & Li, T. (2020). A Novel Image Encryption Approach Based on a Hyperchaotic System, Pixel-Level Filtering with Variable Kernels, and DNA-Level Diffusion. *Entropy, 22*(1), 5.

Xi, H., Yu, S., Zhang, Z., Deng, K., & Xi, H. (2010, 29-31 Oct. 2010). *Generation of Hyperchaotic Chua System via State Feedback Control*. Paper presented at the 2010 International Workshop on Chaos-Fractal Theories and Applications.

Xu, C., Sun, J., & Wang, C. (2020). A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems. *Multimedia Tools and Applications, 79*(9), 5573-5593. doi:10.1007/s11042-019-08273-x

Yaghouti Niyat, A., Moattar, M. H., & Niazi Torshiz, M. (2017). Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Optics and Lasers in Engineering, 90*, 225-237. doi:https://doi.org/10.1016/j.optlaseng.2016.10.019

Yang, F., Mou, J., Liu, J., Ma, C., & Yan, H. (2020). Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application. *Signal Processing, 169*, 107373. doi:https://doi.org/10.1016/j.sigpro.2019.107373

Zeng, J., & Wang, C. (2021). A Novel Hyperchaotic Image Encryption System Based on Particle Swarm Optimization Algorithm and Cellular Automata. *Security and Communication Networks, 2021*, 6675565. doi:10.1155/2021/6675565

Zhang, Q., & Han, J. (2021). A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. *Multimedia Tools and Applications, 80*(9), 13841-13864. doi:10.1007/s11042-020-10437-z

Zhang, Y.-Q., He, Y., Li, P., & Wang, X.-Y. (2020). A new color image encryption scheme based on 2DNLCML system and genetic operations. *Optics and Lasers in Engineering, 128*, 106040. doi:https://doi.org/10.1016/j.optlaseng.2020.106040

Zhou, M., & Wang, C. (2020). A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. *Signal Processing, 171*, 107484. doi:https://doi.org/10.1016/j.sigpro.2020.107484

Zhu, S., & Zhu, C. (2019). Plaintext-Related Image Encryption Algorithm Based on Block Structure and Five-Dimensional Chaotic Map. *IEEE Access, 7*, 147106-147118. doi:10.1109/ACCESS.2019.2946208