International Journal of Safety and Security Engineering
Vol. 15, No. 9, September, 2025, pp. 1933-1940
Journal homepage: http://iieta.org/journals/ijsse

Image Encryption Algorithm Based on a New Four-Dimensional Hyper Chaotic System and Second Order Fuzzy Differential Equations

Zainab Ali Dheyab*, Sadiq A. Mehdi

Department of Computer Science, Mustansiriyah University, Baghdad 10052, Iraq

Corresponding Author Email: zainabali99@uomustansiriyah.edu.iq

https://doi.org/10.18280/ijsse.150916

Received: 24 August 2025
Revised: 24 September 2025
Accepted: 27 September 2025
Available online: 30 September 2025

Keywords:
S-box, second-order fuzzy differential, encryption algorithm, 4D hyper chaotic system, color image

ABSTRACT

Given the ongoing need for data security and the ongoing developments that have led to the transmission of sensitive images in various fields, such as the military, forensic image analysis, and medical, the need arose to design this algorithm, which consists of several encryption steps. Design a new hyper-chaotic four-dimensional (4D) system and use it to create a substitution box (S-box) to do substitution and encrypt the color channels by applying different sequence values with XOR for each channel. Also, a second-order fuzzy differential equation is used to generate the final encryption key, which is then XORed with the original image to produce an encrypted color image, thereby enhancing diffusion and confusion. The algorithm showed high accuracy results in examining tests such as entropy value reach 7.999, Histogram, Number of Pixels Change Rate (NPCR), Peak Signal-to- Noise Ratio (PSNR) above than 99.60 percent, Time speed less than 1.07 sec, Unified Average Changing Intensity (UACI) close to 33 percent, Mean Squared Error (MSE), Correlation coefficient test less than (0.001), for the three directions: vertical, horizontal, and diagonal. Confirms the algorithm's ability to withstand brute-force exploration, statistical analysis, and differential cryptanalysis, ensuring a high level of security and preserving the transmitted image from tampering.

1. INTRODUCTION

The widespread use of multimedia, particularly images, across fields such as geology, biology, medicine, and the military has enabled technological progress that has also enabled attacks, resulting in the theft or falsification of information [1, 2]. The possibility of hiding information within them or preventing the leakage of essential images exists, as does the presence of open networks for communication between people, which expose data to danger. Therefore, it is crucial to maintain the security of information and the safety of images sent over the internet [3-6]. From this standpoint, the focus has shifted to encrypting images and developing algorithms that are highly secure while also restoring them to full resolution without data loss. Scientists and researchers are compelled to build and innovate algorithms to enhance security [7-10]. One of the techniques used is the substitution box (S-box) generated from chaotic systems, which disperses pixel correlations as a preliminary step before encryption [11, 12].

These chaotic systems have initial values that generate non-periodic, sensitive-to-change random numbers, making them a type of complex, non-linear system. Also, multiple steps or rounds can be used for encryption, enhancing security. Also, two stages are used: configuration and diffusion. Encryption is of two types: block encryption, which treats data and images as blocks of bits or bytes, and stream encryption, which treats data bit by bit or byte by byte [13-16].

2. RELATED WORKS

In 2022, Abdallah et al. [17] generated a 16 × 16 S-box using a chaotic map based on a 2D Henon map, thereby enhancing diffusion and minimizing pixel correlation. Also used random sequences to perform the diffusion and configuration steps via X-OR operations, generating random values from the Logistic, Henon, and Lorenz systems.

In 2023, Shakir et al. [18] encrypted a color image using an S-box for substitution and sequences generated from a new four-dimensional (4D) chaotic system that invests the sequence output values, then permuted to eliminate correlations. After it, split the color bands, and for each band, there is a sequence of values to be scrambled. By applying the S-box and utilising other operations, such as DNA, XORing, additions, and subtraction, we can further encrypt the image.

In 2025, Al-Dayel et al. [19] utilised a 4D system to encode color images, leveraging its high randomness, non-linearity, and discrete properties. To enhance these properties, the Langton's Ant was incorporated as a second layer of encoding, which has several rules, and they were used for encoding, as shown in the table in the research paper and the diagram as well, so that these rules lead to unexpected and non-linear patterns, and they are applied iteratively.

3. THE NEW 4D HYPER-CHAOTIC SYSTEM

Create the new 4D hyper chaotic system, a mathematical model containing thirteen parameters, non-linear terms are obtained as:

$$\begin{aligned}\frac{dx}{dt} &= ayz - bx - cw + b\sin(y)\\ \frac{dy}{dt} &= dxz + ex - y - bx\mathrm{Exp}(z)\\ \frac{dz}{dt} &= fxy - gz + hy^2 - i\mathrm{Sin}(w)\\ \frac{dw}{dt} &= jyz + kxz - lw + mz\mathrm{Exp}(x)\end{aligned} \tag{1}$$

where, value of initial conditions are $x(0) = 0.5$, $y(0) = 0.4$, $z(0) = 1.5$, $w(0) = 0.6$, and thirteen initial parameters are $a = 5.8$, $b = 3$, $c = 0.5$, $d = 0.8$, $e = 17$, $f = 29$, $g = 2.4$, $h = 9$, $i = 5$, $j = 15$, $k = 4$, $l = 2.1$ and $m = 2$.

Lyapunov Exponent and fixed points

The Lyapunov Exponent values are $L.E_1 = 2.88108$, $L.E_2 = 0.0828833$, $L.E_3 = -2.48959$, and $L.E_4 = -25.0176$, There are two positive values this mean hyper chaotic system The new hyper chaotic system have two fixed points are E_0 $\{x = 0, y = 0, z = 0, w = 0\}$ and E_1 $\{x = 0.178759, y = 0.266214, z = 1.72831, w = 5.84311\}$, and the new hyper chaotic system instable in fixed point.

Phase portraits

The system exhibits complex, chaotic behavior, as shown in the phase diagram. Figures 1 and 2 show the three and two-dimensional strange attractors. Numerical simulations were performed using Mathematica. The generated shape in the phase resembles the flapping of a butterfly's wings, hence the term "butterfly effect," and because even a small change can cause a significant change in the results.

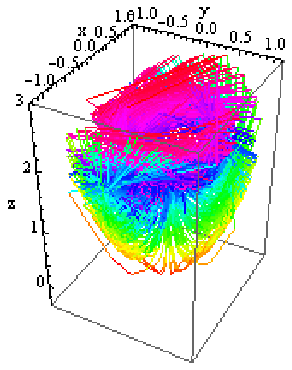


Figure 1. Three-dimensional view of chaotic attractors

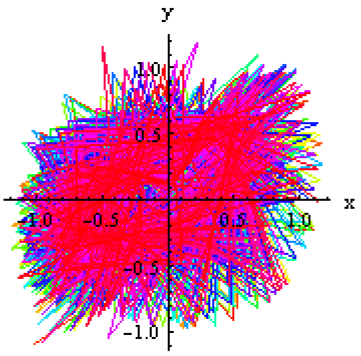


Figure 2. Chaotic attractors of 2D x-y phase

Bifurcation diagram

To understand what happens to a small number of b values within a large range b∈ [4.9, 5]. The bifurcation diagram in Figure 3 shows that as b values increase, a cyclic doubling occurs in the region $4.9 \leq b \leq 5$.

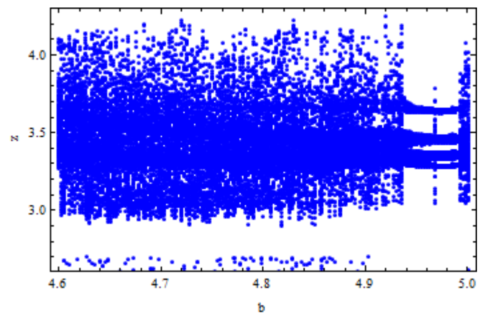


Figure 3. The bifurcation diagram of $b \in [4.9, 5]$

Sensitivity to initial conditions

One of the most important characteristics of a system is its extreme sensitivity to even the slightest change in values. No matter how close the values of two different initial conditions may seem, they will eventually diverge. Therefore, there will come a time when the system's state becomes unpredictable. Figure 4 illustrates that the evolution of the chaotic path is highly sensitive to the initial conditions. The solid line is (x(0) = 0.5, y(0) = 0.4, z(0) = 1.5, w(0) = 0.6) and the value change is (x(0) = 0.5, y(0) = 0.4, z(0) = 1.5, w(0) = 0.60000000000001) for the dashed line.

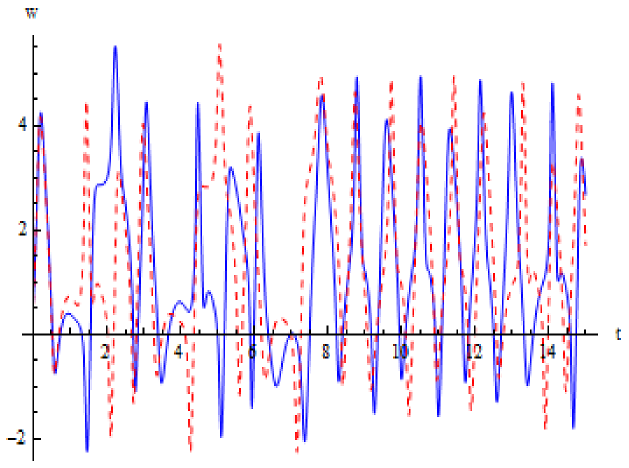


Figure 4. Sensitivity tests of the of the trajectory w(t)

4. PROPOSED ENCRYPTION METHOD

The algorithm uses several encryption phases, each strengthening the encryption. In the first phase, S-boxes are generated using a hyper-chaotic system, while in the second phase, random sequences are generated to encrypt the image using XOR. In the final phase, a key is generated using a second-order fuzzy differential equation. These steps are illustrated through a block diagram in Algorithm 1 and Figure 5. The inverse of these steps represents the decryption algorithm. The decryption process begins with the fuzzy equation key, followed by reversing the XOR operations, and then applying the substitution operation using the S-box, as shown in Algorithm 2 and the block diagram in Figure 6.

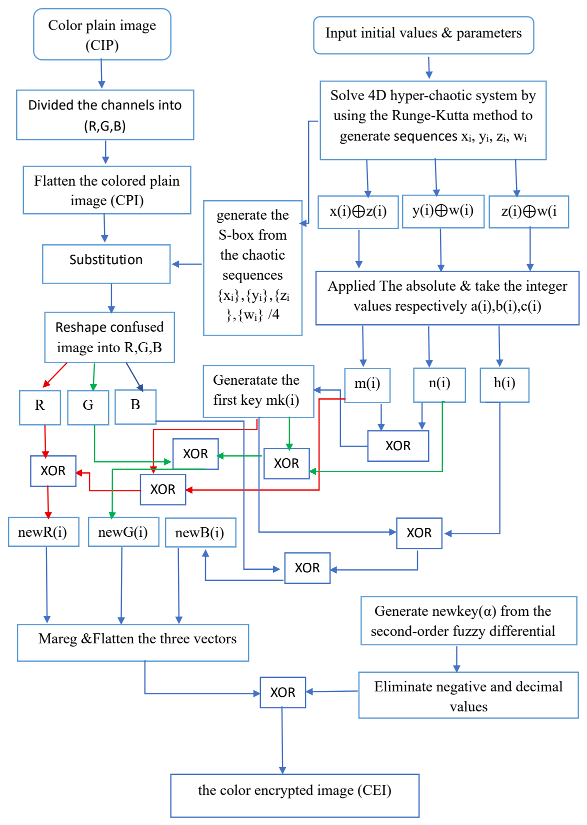
Algorithm 1: Encryption

```
Input:    Color plain image (CPI), parameters: $a, b, c, d, e, f, g, h, i, j, k, l, m$, initial
          $w_0$, $z_0$, $y_0$, $x_0$ from the new hyper chaotic system (1).
Output:   Color encrypt image (CEI).
Step 1:   Read the colored plain image (CPI).
Step 2:   Split the channels into three components (red (R), green (G), and blue (B)).
Step 3:   for i = 0 to image size / 4
          Solve the new hyper-chaotic system (1) by using the Runge-Kutta method to get four chaotic sequences $\{\{x_i\}, \{y_i\}, \{z_i\}, \{w_i\}\}$.
Step 4:   Generate the S-box from the chaotic sequences $\{\{x_i\}, \{y_i\}, \{z_i\}, \{w_i\}\}$by apply the equation below,
          for j = 1 to 256;
Step 5:   $\text{S-box}(j) = (\{x_i\}, \{y_i\}, \{z_i\}, \{w_i\} \times 10^{14}) \bmod 256.$
          Flatten the colored plain image (CPI) and iterate the S-box to operate on each pixel's substitution to get a confused image as
        a confusion state.
          S-box(j) = image
Step 6:   Generate three sequence values from the new hyper chaotic system (1)
          for i = 0 to image size,
        $a(i) = x(i) \oplus z(i),$
        $b(i) = y(i) \oplus w(i),$
        $c(i) = z(i) \oplus w(i).$
Step 7:   Generate the first key from the sequences,
          for i = 0 to image_size,ol
        $m(i) = \text{abs}(a(i) \times 10^{14}) \bmod 256,$
        $n(i) = \text{abs}(b(i) \times 10^{14}) \bmod 256,$
        $h(i) = \text{abs}(c(i) \times 10^{14}) \bmod 256,$
Step 8:   Diffuse the pixels of the confusion state by using a key generated from Step 6 that matches the input image size
          for i = 0 to image_size
        $\left(mk(i) \oplus m(i)\right) \oplus R(i) = newR(i),$
        $(mk(i) \oplus n(i)) \oplus G(i) = newG(i),$
        $(mk(i) \oplus h(i)) \oplus B(i) = newB(i).$
Step 9:   Combine the three vectors (newR(i), newG(i), newB(i)) to obtain the first encrypted image (FEI).
Step 10:  Use the first encrypted image (FEI) generated in the previous steps to apply a second state of encryption.
Step 11:  Generate a second key from the second-order fuzzy differential equation,
          for α = 0 to 1 step 100001, x = 0.5,
          $key(\alpha) = \left((1 - \alpha)\left(1 + x - \frac{1}{2}x^2\right)\right)$
Step 12:  Eliminate negative and decimal values from the resulting values of the second-order fuzzy differential equation,
          $newkey\ (\alpha) = \text{abs}(key(\alpha) \times 10^{14})\bmod 256.$
Step 13:  Apply the XOR on newkey with the first encrypted image (FEI),
          $newkey(\alpha) \oplus (FEI) = CEI.$
Step 14:  Save the color encrypted image (CEI).
Step 15:  End.
```

Algorithm 2: Decryption

```
Input:    color encrypt image (CEI) and the second key.
Output:   color decrypt image (CDI).
Step 1:   Read the color encrypt image (CEI).
Step 2:   Generate a second key from the second-order fuzzy
          differential equation
          for α = 0 to 1 step 100001, x = 0.5,
          $key(\alpha) = \left((1 - \alpha)\left(1 + x - \frac{1}{2}x^2\right)\right)$
Step 3:   Eliminate negative and decimal values from the
          resulting values of the second-order fuzzy differential equation,
          newkey (α) = abs(key(α) * 10^14) mod 256.
Step 4:   Apply the XOR on newkey with the color encrypt image (CEI) to get the first decrypted image (FDI),
          $newkey(\alpha) \oplus (CEI) = FDI.$
Step 5:   Use the first decrypted image (FDI) generated in the previous steps to apply a second state of decryption.
Step 6:   Split the channels of the first decrypted image (FDI) into three vectors (newR(i), newG(i), newB(i)).
Step 7:   for i = 0 to image size,
          Solve the new hyper-chaotic system (1) by using the Runge-Kutta method to get four chaotic sequences {{xi},{yi},{zi},{wi}}.
Step 8:   Generate three sequence values from the new hyper chaotic system (3.1)
          $a(i) = x(i)\oplus z(i),$
          $b(i) = y(i)\oplus w(i),$
          $c(i) = z(i)\oplus w(i).$
          Generate the first key from the sequences,
Step 9:   for i = 0 to image_size ,
          $m(i) = \text{abs}( a(i) *10^{14}) \bmod 256,$
          $n(i) = \text{abs}( b(i) *10^{14}) \bmod 256,$
          $h(i) = \text{abs}( c(i)*10^{14}) \bmod 256,$
          $mk(i) = m(i) \oplus n(i).$
```

```
Step 10:  Diffuse the pixels of the confusion state by using
          a key generated from Step 9 that matches the input image size
          for i = 0 to image_size
          $R(i) = newR(i) \oplus (mk(i) \oplus m(i))$,
          $G(i) = newG(i) \oplus (mk(i) \oplus n(i))$,
          $B(i) = newB(i) \oplus (mk(i) \oplus h(i))$.
Step 11:  Use the first decrypted image (FDI) that was generated
          in the previous steps to apply the final step of substitution.
Step 12:  Generate the inverse S-box,
          for j = 1 to 256;
          $S\text{-box-}1(j) = (\{xi\}, \{yi\}, \{zi\}, \{wi\}*10^{14}) \bmod 256$.
Step 13:  Iterates the inverse S-box to cover the Flatten image
          that operates on each pixel's substitution to get the
          color decrypted image (CDI),
          S-box-1(j) = image
Step 14:  Save the color decrypted image (CDI).
Step 15:  End.
```


Color plain image (CIP)
Input initial values & parameters
Divided the channels into (R,G,B)
Solve 4D hyper-chaotic system by using the Runge-Kutta method to generate sequences x_i, y_i, z_i, w_i
Flatten the colored plain image (CPI)
generate the S-box from the chaotic sequences {xi},{yi},{zi},{wi} /4
x(i)⊕z(i)
y(i)⊕w(i)
z(i)⊕w(i
Substitution
Applied The absolute & take the integer values respectively a(i),b(i),c(i)
Reshape confused image into R,G,B
Generatate the first key mk(i)
m(i)
n(i)
h(i)
R
G
B
XOR
newR(i)
newG(i)
newB(i)
Mareg &Flatten the three vectors
Generate newkey(α) from the second-order fuzzy differential
Eliminate negative and decimal values
the color encrypted image (CEI)

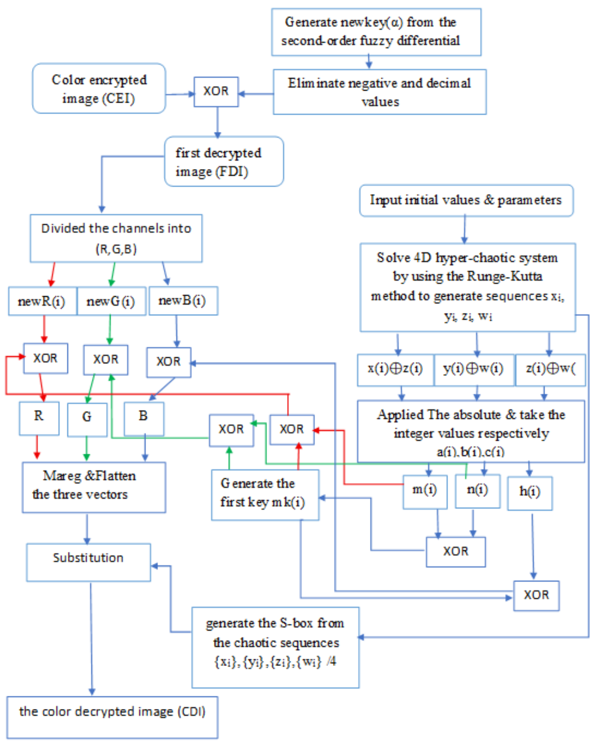

Generate newkey(α) from the second-order fuzzy differential
Color encrypted image (CEI)
XOR
Eliminate negative and decimal values
first decrypted image (FDI)
Input initial values & parameters
Divided the channels into (R,G,B)
Solve 4D hyper-chaotic system by using the Runge-Kutta method to generate sequences x_i, y_i, z_i, w_i
newR(i)
newG(i)
newB(i)
x(i)⊕z(i)
y(i)⊕w(i)
z(i)⊕w(
R
G
B
Applied The absolute & take the integer values respectively a(i),b(i),c(i)
Mareg &Flatten the three vectors
Generate the first key mk(i)
m(i)
n(i)
h(i)
Substitution
generate the S-box from the chaotic sequences {xi},{yi},{zi},{wi} /4
the color decrypted image (CDI)

Figure 5. The three-phase encryption algorithm block diagram

Figure 6. The three-phase decryption algorithm block diagram

Table 1. S-box is generated from the new 4D hyper-chaotic system

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	64	78	139	185	120	183	9	212	71	15	174	118	8	75	193	55
1	162	40	49	111	186	175	236	217	198	90	27	241	134	19	223	13
2	85	169	155	50	67	249	91	205	202	157	105	70	150	3	206	46
3	144	211	204	133	79	106	224	84	176	87	61	4	188	167	48	41
4	45	30	243	146	137	251	44	165	96	154	131	108	179	191	252	195
5	18	159	253	227	97	47	192	69	170	247	141	219	80	215	218	245
6	92	66	238	199	37	1	102	32	31	177	233	187	234	38	20	56
7	7	152	126	5	200	231	43	81	22	158	82	148	254	128	136	73
8	60	99	147	14	196	52	220	109	164	104	98	171	39	16	140	29
9	190	122	77	10	239	149	65	72	184	11	142	235	222	160	83	166
A	124	242	101	225	100	230	208	58	125	180	114	143	0	178	68	153
B	2	88	26	246	21	94	240	132	173	129	145	255	214	209	63	229
C	103	207	248	161	42	110	203	172	53	135	116	117	197	194	12	115
D	95	107	163	228	156	182	25	93	24	244	123	35	86	74	168	237
E	151	36	232	201	226	121	213	17	62	250	119	57	33	138	51	59
F	89	181	34	112	54	130	28	113	23	210	127	6	76	221	189	216

4.1 Dynamic S-box phase based on the new 4D hyper-chaotic system

From the new 4D hyper-chaotic system, which is solved using the Runge-Kutta method, a new S-box, as shown in Table 1, is built to be applied to the image before the two encryption phases. The size of the new S-box (16 × 16) provides 256 unique values for substitution. The substitution is applied to all three channels (red, green, blue). The robustness and sensitivity of the hyper-chaotic system (1) are directly proportional to the robustness and sensitivity of the S-box.

4.2 Encryption second phase based on the new 4D hyper-chaotic system

In this phase, new values are assigned by performing an XOR between the various system sequences. Generate a key from the sequences, then XOR it with the new values generated by the new hyper-chaotic system, and apply the result to the image after applying the S-box, as described in the encryption algorithm and block diagram.

4.3 Encryption final phase based on the second-order fuzzy differential equation

As the final phase, apply the second-order fuzzy differential equation to generate a second key, then perform the XOR operation. The fuzzy equation selects fuzzy input values that change constantly during this step, for α = 0 to 1, with a step size of 100001. These steps explain how the equation was generated. Table 2 shows NIST test was performed on the key generated from this equation, and it passed all tests.

Table 2. NIST test result for the sequence generated by a fuzzy

NIST Test Name	P-Value of Concatenate	Final Results
Frequency (Monobit)	0.3173105078	√
Frequency-test within -a Block (m = 128)	0.4795001221	√
Run	0.4795225386	√
(Longest-run) of ones in a block	0.254925402280664	√
Rank of Binary-matrix	0.0391046157	√
Discrete-Fourier-Transformation	0.6463551955	√
Non-Overlapping Template	0.0381950649	√
Overlapping Template	0.4328750525	√
Linear Complexity Test	0.2519033436	√
Serial-1	0.9999999496	√
Serial-2	0.9999947013	√
Approximate Entropy	0.8886599723	√
Cumulative sums	0.54073062317	√
Random Excursion	0.0156999496	√
Random Excursions Variant	0.0030673968	√

$$\frac{\partial^2 \widetilde{w}(x,y)}{\partial x^2} = \beta,$$
$$\widetilde{w}(x,0) = \frac{\partial \widetilde{w}(x,0)}{\partial y} = \left[\underline{w}(0,y,\alpha), \underline{w}(0,y,\alpha)\right] = [\alpha - 1, 1 - \alpha],$$
$$\beta = [\alpha - 1, 1 - \alpha], \ \forall \alpha \in [0,1].$$

$\widetilde{w}(x,y)$ is differentiable and $\frac{\partial \widetilde{w}(x,y)}{\partial x}$ is differentiable:

$$\underline{w}(x,y,\alpha) = (1-\alpha)\left(1 + x - \tfrac{1}{2}x^2\right), \underline{w}(x,y,\alpha) = (\alpha - 1)\left(1 + x - \tfrac{1}{2}x^2\right)$$

for α = 0 to 1 step 100001, x = 0.5,

$$key(\alpha) = \left((1 - \alpha)\left(1 + x - \tfrac{1}{2}x^2\right)\right) \quad (2)$$

5. EVALUATION OF RESULTS AND SECURITY ANALYSIS

Using different color image diminution with varying types of extensions, the Lake image (1024 × 1024 .jpg file), boat image (512 × 512 .png file), and Peppers image (256 × 256 .tiff file) were tested. The algorithm was simulated using Python 3.9.7-amd64 and Visual Studio 2022. The numerous tests conducted yielded excellent results, including a histogram analysis. The results demonstrated that this method is robust, as evidenced by its appearance in various attack analyses, including Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). It is difficult to hack or break, making it highly suitable for encrypting color images. Also, when decrypting the generated image, it matches the original one exactly, with no errors or missing values, as shown in the Peak Signal-to- Noise Ratio (PSNR) and Mean Squared Error (MSE).

5.1 Histogram analysis

Table 3 illustrates histogram analysis, a crucial statistic for measuring encryption quality. We observe that images in section b show increases and decreases in frequency, representing the repetitions within the image that reveal its content. In contrast, images in section d exhibit a stable distribution, lacking statistical similarity in visual appearance.

5.2 Information entropy analysis

In Table 4, a higher entropy value, close to 8, is obtained using the Shannon entropy measure, indicating higher security when calculating the frequency of image pixels. It is a perfect result that repels statistical attacks. The entropy image value is shown in the table below.

Table 3. Encryption results (a) org-imgs, (b) enc-imgs, (c) enc-histogram, (d) dec-imgs, (e) dec-histogram

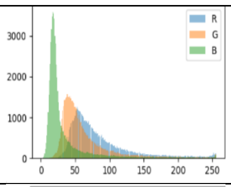
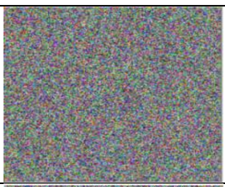
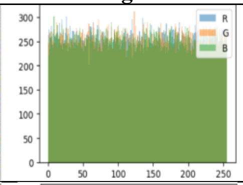
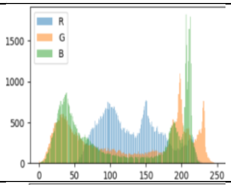
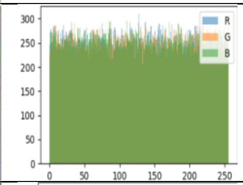
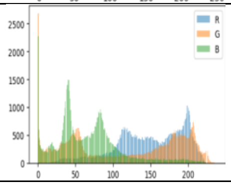
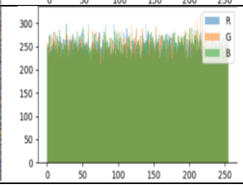
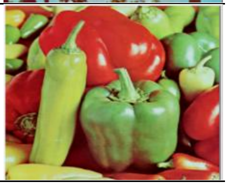
Original image	Size	Extension	Original Histogram	Encrypted	Encrypted Histogram	Decrypted
	1024 × 1024	.jpg				
	512 × 512	.png				
	256 × 256	.tiff				
(a)			(b)	(c)	(d)	(e)

Table 4. Entropy of encrypted and original images

Image Type	Entropy Original	Entropy Encrypted
Peppers 256 × 256.tiff	7.7031	7.9990
Sailboat 512 × 512.png	7.7393	7.9990
Lake 1024 × 1024.jpg	7.1197	7.9991

5.3 Correlation coefficient analysis

A coefficient value approaching zero indicates stronger encryption performance, as it implies minimal statistical similarity between the two images. In such cases, the encrypted image lacks discernible visual patterns, making it extremely difficult to interpret or reconstruct without possession of the correct decryption key [20]. Table 5 presents the results evaluating the robustness of the proposed image encryption method, explicitly using the correlation coefficient between the encrypted and original images.

Table 5. Correlation coefficient test for the three directions: (V, H, and D) for the encrypted and original images of the three images 'Peppers', 'Sailboat', and 'Lake'

Image	Original			Encrypted		
	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
Peppers	0.9712	0.9643	0.9381	0.0037	0.0005	-0.0021
Sailboat	0.9568	0.9583	0.9288	0.0026	0.0004	-0.0044
Lake	0.9000	0.9022	0.8407	-0.0020	0.0041	-0.0010

5.4 Security differential attack test

One important test for evaluating the algorithm's efficiency is differential attack testing. The algorithm must prevent attacks on the encryption. The algorithm must ensure that any slight change in the original image results in a change in the encrypted image. This is measured using NPCR, where accuracy is above 99%, and UACI, where it is around 33%. Table 6 shows the results of the differential tests.

Table 6. NPCR and UACI results

Name	Size	NPCR	UACI
Peppers	256 × 256	99.60%	32.11%
Sailboat	512 × 512	99.62%	31.96%
Lake	1024 × 1024	99.61%	35.07%

5.5 Execution and complexity analysis of time

Table 7 contains the execution times for encrypting and decrypting the image, and the values are off by 1 second or a few parts. When using a specific encryption algorithm or designing a new one, time is one of the most important factors to consider. Therefore, based on previous tests that demonstrated its strength and the inability to decode it, the encryption time values clearly support the findings of the research. used Python 3.9, Visual Basic 2022, 16GB of RAM, and Windows 11. Table 8 illustrates the time allocation for each operation in the algorithm. The execution time increases with increasing input size and the number of pixels in the image. The time complexity depends entirely on the image size and the operations performed on the image during encryption to generate keys and perform the encryption.

Table 7. Time speed of three different image sizes

Image	Size	Encryption Time in sec.	Decryption Time in sec.
Peppers	256 × 256	1.0402 sec	0.9794 sec
Sailboat	512 × 512	1.0709 sec	1.0521 sec
Lake	1024 × 1024	1.0340 sec	1.0075 sec

Table 8. The execution time of the algorithm

Process	Time [unit: sec]
Chaotic system keys	0.9578
S-box	0.00014
First encryption	0.283692
Second encryption	0.24491

5.6 Analysis of MSE and PSNR ratio

Two essential metrics for evaluating decryption quality are PSNR and MSE. The current algorithm returns zero, indicating a perfect match between the decrypted image and the original, as shown in Table 9.

5.7 Key space analysis

One of the important properties of algorithms is the size of the key space. The key space represents the number of possible keys generated in the system. The larger the space, the greater the ability to repel attacks, as the key used cannot be predicted. Based on the system's equations, the space is $(10^{14})^{17} = 10^{238} \simeq 2^{757}$, which is larger than (2^{128}), making it resistant to brute-force attacks.

5.8 Comparison results with other encryption methods

In Table 10, the work is compared with previously published works [2, 18], where better results were obtained, especially in entropy, as well as differences in the values reported in the previous studies [2, 18]. The Sailboat image was initially used at a small size, but in this work, it was used at 512 × 512. This resulted in better performance in terms of Entropy, NPCR, and UACI, with close correlations. As for comparing the Pepper image with the study by Shakir et al. [18], it is the same size, but in this work, it is colored, which means more complexity, and the results also show our work, especially in entropy.

Table 9. PSNR and MSE for plain and decrypted images

Image	Size	PSNR	MSE
Peppers	256 × 256	Inf dB	0.000
Sailboat	512 × 512	Inf dB	0.000
Lake	1024 × 1024	Inf dB	0.000

Table 10. Proposed algorithm comparison results with other encryption methods for the Sailboat and Pepper images

Image	Size	Entropy	NPCR	UACI	Original			Encrypted		
					V	H	D	V	H	D
Sailboat ours	512 × 512	7.9990	99.62%	31.96%	0.9568	0.9583	0.9288	0.0026	0.0004	-0.0044
Ref. [2]	256 × 256	7.9975	99.60%	33.4%	-	-	-	-	-	-
Ref. [18]	256 × 256	7.9989	99.61%	34.02%	0.9644	0.9679	0.9504	-0.0012	-0.0012	-0.0009
Pepper ours	256 × 256 color	7.9990	99.60%	32.11%	0.9712	0.9643	0.9381	0.0037	0.0005	-0.0021
Ref. [2]	256 × 256 gray	7.9974	99.60%	33.46%	0.9654	0.9680	0.9443	0.0003	0.0005	-0.0400

6. CONCLUSIONS

A new algorithm was generated for a 4D hyperchaotic system, which provides robust encryption. An S-box was generated from the new system. A key was generated from the fuzzy logic equation, yielding optimal results. The correlation coefficient and perfect values of entropy, NPRC, and UACI were all close to perfection. The tables in the research also show many values obtained by encrypting a set of images with different extensions and sizes. It uses a high-strength algorithm, making it difficult to decrypt or hack. In future work, consider incorporating artificial intelligence methods to reduce the time.

ACKNOWLEDGMENT

This work gratefully acknowledges the support and assistance provided by Mustansiriyah University (https://www.uomustansiriyah.edu.iq) which played a vital role in the completion of this work.

REFERENCES

[1] Jasim, S.H., Hoomod, H.K., Hussein, K.A. (2024). Image encryption based on hybrid parallel algorithm: DES-present using 2D-chaotic system. International Journal of Safety and Security Engineering, 14(2): 633-646. https://doi.org/10.18280/ijsse.140229

[2] Sheng, Y., Li, J., Di, X., Li, X., Xu, R. (2022). An image encryption algorithm based on complex network scrambling and multi-directional diffusion. Entropy, 24(9): 1247. https://doi.org/10.3390/e24091247

[3] Yan, S., Li, L., Zhao, W., Gu, B. (2023). Design of a new four-dimensional chaotic system and its application to color image encryption. Non-linear Dynamics, 111(18): 17519-17545. https://doi.org/10.1007/s11071-023-08726-x

[4] Shakir, H.R., Mehdi, S.A., Hattab, A.A. (2023). A new method for color image encryption using chaotic system and DNA encoding. Mustansiriyah Journal of Pure and Applied Sciences, 1(1): 68-79. https://doi.org/10.47831/mjpas.v1i1.9

[5] Enayatifar, R., Abdullah, A.H., Lee, M. (2013). A weighted discrete imperialist competitive algorithm (WDICA) combined with chaotic map for image encryption. Optics and Lasers in Engineering, 51(9): 1066-1077. https://doi.org/10.1016/j.optlaseng.2013.03.010

[6] Zghair, H.K., Mehdi, S.A., Sadkhan, S.B. (2020). Design and analytic of a novel seven-dimension hyper chaotic systems. Proceedings of the 2020 1st Information Technology to Enhance e-Learning and Other Application (IT-ELA), Baghdad, Iraq, pp. 77-81.

http://doi.org/10.1109/IT-ELA50150.2020.9253077

[7] Zghair, H.K., Mehdi, S.A., Sadkhan, S.B. (2021). Speech scrambler based on discrete cosine transform and novel seven-dimension hyper chaotic system. Journal of Physics: Conference Series, 1804(1): 012048. https://doi.org/10.1088/1742-6596/1804/1/012048.

[8] Alghamdi, Y., Munir, A., Ahmad, J. (2022). A lightweight image encryption algorithm based on chaotic map and random substitution. Entropy, 24(10): 1344. https://doi.org/10.3390/e24101344

[9] Kuffi, E.A., Mehdi, S.A., Mansour, E.A. (2022). Color image encryption based on new integral transform SEE. Journal of Physics: Conference Series, 2322(1): 012016. http://doi.org/10.1088/1742-6596/2322/1/012016

[10] Alshekly, T.K., Albahrani, E.A., Lafta, S.H. (2022). 4D chaotic system as random substitution-box. Multimedia Tools and Applications, 81(11): 15793-15814. http://doi.org/10.1007/s11042-022-11928-x

[11] Zghair, H.K., Mehdi, S.A., Sadkhan, S.B. (2020). Analysis of novel seven-dimension hyper chaotic by using SDIC and waveform. In Proceedings of the 2020 3rd International Conference on Engineering Technology and Applications (IICETA), Najaf, Iraq, pp. 95-99. http://doi.org/10.1109/IICETA50496.2020.9318940

[12] Dimitrov, M.M. (2020). On the design of chaos-based S-boxes. IEEE Access, 8: 117173-117181. https://doi.org/10.1109/ACCESS.2020.3004526

[13] Jasem, N.N., Mehdi, S.A. (2023). Multiple random keys for image encryption based on sensitivity of a new 6D chaotic system. International Journal of Intelligent Engineering and Systems, 16(5): 576-585. http://doi.org/10.22266/ijies2023.1031.49

[14] Al-Maadeed, T.A., Hussain, I., Anees, A., Mustafa, M.T. (2021). An image encryption algorithm based on chaotic Lorenz system and novel primitive polynomial S-boxes. Multimedia Tools and Applications, 80(16): 24801-24822. https://link.springer.com/article/10.1007/s11042-021-10695-5

[15] Abduljabbar, Z.A., Abduljaleel, I.Q., Ma, J., Al Sibahee, M.A., Nyangaresi, V.O., Honi, D.G., Abdulsada, A.I., Jiao, X. (2022). Provably secure and fast color image encryption algorithm based on S-boxes and hyperchaotic map. IEEE Access, 10: 26257-26270. https://doi.org/10.1109/ACCESS.2022.3151174

[16] Alexan, W., Elkandoz, M., Mashaly, M., Azab, E., Aboshousha, A. (2023). Color image encryption through chaos and KAA map. IEEE Access, 11: 11541-11554. https://doi.org/10.1109/ACCESS.2023.3242311

[17] Abdallah, A.A., Farhan, A.K. (2022). A new image encryption algorithm based on multi chaotic system. Iraqi Journal of Science, 63(1): 324-337. https://doi.org/10.24996/ijs.2022.63.1.31

[18] Shakir, H.R., Mehdi, S.A., Hattab, A.A. (2023). A new four-dimensional hyper-chaotic system for image encryption. International Journal of Electrical and Computer Engineering, 13(2): 1744-1756. http://doi.org/10.11591/ijece.v13i2.pp1744-1756

[19] Al-Dayel, I., Nadeem, M.F., Khan, M.A., Abraha, B.S. (2025). An image encryption scheme using 4-D chaotic system and cellular automaton. Scientific Reports, 15(1): 19499. https://doi.org/10.1038/s41598-025-95511-y

[20] Jasim, O.A., Amer, S.R., Hussein, S.F., Mehdi, S.A. (2024). Enhanced image encryption using a novel chaotic system and scramble dithering technique. International Journal of Safety & Security Engineering, 14(5): 1465-1476. https://doi.org/10.18280/ijsse.140514