INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Image Encryption Algorithm with Block Scrambling Based on Logistic Map

M Y Shiju Thomas[1,2]*, **V N Addapalli Krishna**[1], **M Bindiya Varghese**[2]

1 Computer Science and Engineering, School of Engineering and Technology, Christ University, Bangalore, Karnataka, India
2 Rajagiri College of Social Sciences (Autonomous), Kerala, India

OPEN ACCESS

Received: 26-04-2022

Accepted: 04-03-2023

Published: 06-04-2023

Citation: Thomas MYS, Krishna VNA, Varghese MB (2023) Image Encryption Algorithm with Block Scrambling Based on Logistic Map. Indian Journal of Science and Technology 16(14): 1045-1055. https://doi.org/10.17485/IJST/v16i14.900

* **Corresponding author**.

shijuthomascochin@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To develop an efficient algorithm based on block-scrambling and chaotic maps to generate an encrypted image capable of protecting sensitive information. **Methods:** The proposed technique introduces the two-pass Block-XOR operation column-wise and then swaps pixels within each block in an anti-clockwise crisscross pattern to separate and rearrange nearby pixels into distinct rows and columns. The strong correlations between nearby pixels are disrupted, making it challenging to identify the original image. Afterward, image diffusion using binarization is performed on the obtained scrambled image using a chaotic sequence generated by a logistic map. **Findings:** This scheme has accomplished an efficient encryption method based on block-scrambling and a chaotic map. The experimental results show that the system is competent for generating cipher text by encrypting images and can withstand malicious attacks and thus protect sensitive information. The histogram analysis results show that the proposed system displays adequate resistance to statistical attacks. The histogram of cipher text is uniformly distributed and bears no resemblance to the plaintext histogram. The differential analysis shows that the NPCR and UACI values are greater than 99 and 33, respectively, and are acceptable to conclude that the system is also resistant to differential attacks. The proposed technique experimented with using standard test images from the literature and the USC-SIPI image dataset. **Novelty:** A two-pass Block-XOR operation strengthens the scrambling by altering the value of the image pixel. Then an anti-clockwise crisscross pixel swapping algorithm is utilized for block scrambling.

Keywords: Cryptography; Chaotic Maps; Pixel Confusion; Histogram Analysis; Cyber Attacks

1 Introduction

Digital images are now being transmitted[1] over the internet at high speed and in huge volumes. Given the massive amount of data passed over the network and the type of information it contains, it is apparent that there is a need to ensure the security of such data via appropriate methods. It led to the development of image encryption as a broad area of research. Over the years, several encryption techniques that modify or rearrange the pixel values have been mastered to achieve high confusion and diffusion levels. Chaos-based[2] encryption is highly popular, involves the generation of chaotic sequences for encryption, and has applications in various areas such as communication, the military, healthcare, and marine research. In general, chaos-based encryption techniques involve pixel scrambling and pixel diffusion. The study[3] focuses on a chaotic-based method that uses wavelet transforms and chaotic map properties. This algorithm uses a two-stage encryption process. First, it carried out the picture diffusion process. Additionally, hyper-chaotic sequences significantly lowered the calculation amount in confusion using the wavelet transform. Enhanced Logistic Map (ELM)[4] that is resistant to attacks by using chaotic maps and simple encryption techniques such as block scrambling and modified zigzag transformation for encryption phases, as well as permutation, diffusion, and critical stream generation. Pixel scrambling causes disorientation and dispersion. According to the testing findings, the proposed approach created encrypted images with a uniform distribution of pixel histograms. The study[5] discussed bifurcation in-depth, focusing on general chaos theory and how it can be applied in the real world. The technique demonstrates how chaos theory can generate a random number in cryptography.

Diffusion and confusion are the two crucial phases of image encryption. One or both should be used to develop a trustworthy ciphering system. Asymmetric encryption based on a quantum logistic map and symmetric S-box-based chaotic systems is implemented in[6] and[7], respectively. An effective image-specific chaos-based encryption[8] scheme that consists of a system of two independent chaotic functions with exceptional sensitivity to initial states is used to apply confusion and diffusion concepts to images of any entropy. The first function shuffles pixel positions, while the second changes pixel values. Because of the pixel organisation change, adjacent pixels with naturally relative values will have significantly different values, making it harder to crack the encrypted image. Logical operations like exclusive-or and circular rotation are used to create a system more resistant to differential attacks. A combined zigzag-based distribution and one-dimensional logistical chaotic encryption[9] techniques are applied to images by dividing the image into bit levels. Experiments are done with a one-dimensional logistic self-embedding chaotic system to make the chaotic sequences for the encryption process. Pixel scrambling[10] deals with rearranging the pixels to render the original image unreadable and break the correlations among the neighbouring pixels. In contrast, pixel diffusion involves changing the pixel values in a specific manner to protect image information.

In recent years, several chaos-based encryption methods have been proposed. A gray-level[11] image encryption scheme has been mastered based on permutation and diffusion on image pixels. By putting together a tent map framework and a deterministic finite state machine model, the study[12] makes a new chaotic map. The proposed system[13] scrambles image blocks using a zigzag pattern, rotation, and random permutation, finally diffused by using a key generated by a chaotic logistic map. Chaos-based image encryption systems use two phases of scrambling and spreading to get the encrypted cipher image. The study mastered A 2-dimensional[14] chaotic map synthesised from sine and logistic maps. The proposed method is based on permutation and substitution using a single substitution-box algorithm, and the results show that the proposed technique is efficient and resistant to attacks. An improved sinusoidal chaotic system[15] generates the key for enhanced security. After going through a synchronous shuffling diffusion operation and a dynamic bit-shifting recombination operation, the image becomes encrypted.[16] has perfected an iterative fractal sorting matrix for chaotic picture encryption. Improved logistic maps are applied based on double perturbation and feedback control[17] for the encryption and decryption processes. An analysis of several studies found that low-dimensional chaotic systems are simple to implement since they have only one control parameter. Experimental results show that the proposed algorithm is efficient for practical applications. In the proposed scheme, first, pixel scrambling is applied for reducing the correlation between pixels in the image and afterwards the scrambled image is encrypted using the chaotic sequence generated form the logistic map.

2 Methodology

The block diagram of the mastered scrambling and encryption scheme is depicted in Figure 1. In the proposed scheme, image encryption takes place using the concepts of block scrambling and chaotic maps, in which an image is first scrambled using block scrambling techniques and then encrypted using keys generated by a logistic map algorithm. The obtained encrypted image, called a cipher, can be decrypted back to its original form using reverse scrambling and decryption. A dual iterative block scrambling algorithm named Block-XOR is introduced in this study, and it is explained in detail in Section 2.1.

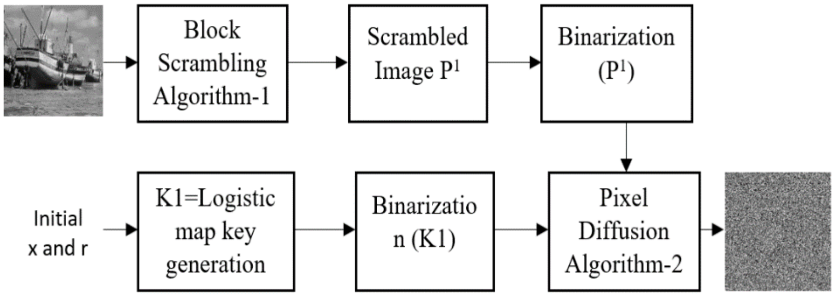
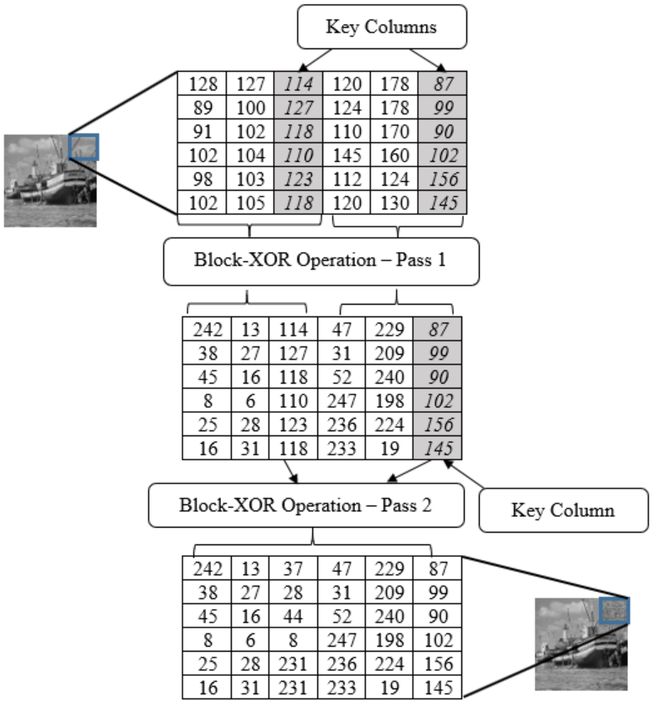

Block Scrambling Algorithm-1
Scrambled Image P1
Binarization (P1)
Initial x and r
K1=Logistic map key generation
Binarizatio n (K1)
Pixel Diffusion Algorithm-2

Fig 1. Proposed scrambling and encryption scheme

2.1 Block-XOR Operation



128	127	*114*	120	178	*87*
89	100	*127*	124	178	*99*
91	102	*118*	110	170	*90*
102	104	*110*	145	160	*102*
98	103	*123*	112	124	*156*
102	105	*118*	120	130	*145*

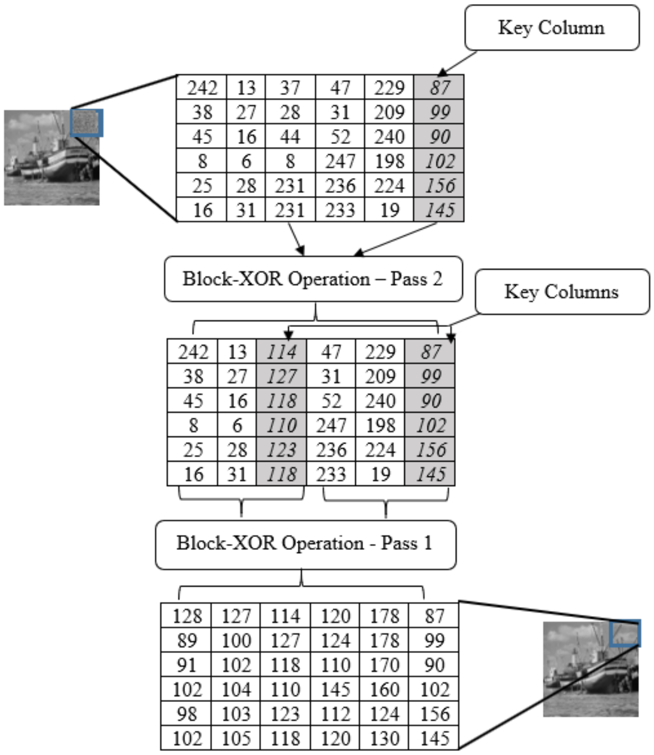
242	13	114	47	229	*87*
38	27	127	31	209	*99*
45	16	118	52	240	*90*
8	6	110	247	198	*102*
25	28	123	236	224	*156*
16	31	118	233	19	*145*

242	13	37	47	229	87
38	27	28	31	209	99
45	16	44	52	240	90
8	6	8	247	198	102
25	28	231	236	224	156
16	31	231	233	19	145

Fig 2. Two-passBlock-XOR Operation

Block-based scrambling is a technique used to separate and rearrange neighboring pixels into different rows and columns such that the strong correlations between adjacent pixels are broken, making it challenging to identify the original structure of the image. The source image is converted to a pixel array using appropriate operations. The introduced two-pass Block-XOR procedure can be applied row-wise or column-wise.

First, a two-pass block scrambling algorithm named Block-XOR is performed on the pixel array column-wise, as shown in Figure 2 . The pixel array is partitioned into data and key columns during the initial iteration. It is often written as a set of M data columns and one key column. In a 512 X 512 image, there are 4 * 128 column blocks, i.e., (3 + 1) * 128, where 3 and 1 represent the number of data columns and key columns, respectively. An XOR operation is applied between the data and key column values. In Figure 2, the Block-XOR algorithm considers a portion of the 6x6-pixel area of the input image. The chosen pixels are divided into 3 x 2 column blocks, or (2+1) x 2 column blocks, each of which has two data columns and one key column. The italic and highlighted columns are the key columns, and the rest of the columns are data columns. The output of pass-1 XOR operations shows that all the data column values have become ciphers.



242	13	37	47	229	*87*
38	27	28	31	209	*99*
45	16	44	52	240	*90*
8	6	8	247	198	*102*
25	28	231	236	224	*156*
16	31	231	233	19	*145*

242	13	*114*	47	229	*87*
38	27	*127*	31	209	*99*
45	16	*118*	52	240	*90*
8	6	*110*	247	198	*102*
25	28	*123*	236	224	*156*
16	31	*118*	233	19	*145*

128	127	114	120	178	87
89	100	127	124	178	99
91	102	118	110	170	90
102	104	110	145	160	102
98	103	123	112	124	156
102	105	118	120	130	145

Fig 3. Two-passBlock-XOR reverse operation

In pass-2, the same procedure is applied to the key columns by considering any key column as the key column. Figure 3 explains the reverse Block-XOR operation. The second stage of block scrambling is swapping the pixels in an anti-clockwise crisscross manner within each block, which is explained in Algorithm 1. The objective is to break the correlations between adjacent pixels and obtain an image that does not resemble the original. Thus, getting the input image without the corresponding unscrambling method is impossible.

Algorithm 1: Image Scrambling

Input: Image P

Output: Scrambled image P^1

```
begin
width, height = size(P);
xres=width;
yres=height;
BLKSZ=width/2
for i=2 to BLKSZ+1 do
    for j=0 to (xres / i) do
        for k=0 to (yres / j)
            rot (arr, i, j*i, k*i) //function to rotate array
        end
    end
end
for i=3 to BLKSZ+1 do
    for j=0 to (BLKSZ+2-i) do
        for k=0 to (BLKSZ+2-i)
            rot (arr, BLKSZ+2-i, j* BLKSZ+2-i, k* BLKSZ+2-i)
        end
    end
```

```
end
P^1 = arr;
return P^1;
end
```

2.2 Encryption

The image obtained from the scrambling process is converted into a pixel array, then encrypted using a key generated by a logistic map. The key is a chaotic sequence produced by the logistic map, converted to a binary sequence. The binarization of the image array forms the binary sequence on the image, upon which a bitwise XOR operation is applied between each pixel and its corresponding key value. The encrypted image is generated from the binary sequence by converting it into an integer value. The encryption process is depicted by Algorithm 2.

Algorithm 2: Image Encryption with Logistic Map

```
Input: Scrambled image P^1, x, r
Output: Encrypted image E
begin
N=row*col
Initialize all cells of matrix x with 0.5
for i=1 to N-1 do
x[n+1] = r*x[n]*(1-x[n])
end
y=x
p=y*255
z=p
Initialize e_arr[row][col] with zeroes
for i=1 to row-1
for j=1 to col-1
p=decimalToBinary (arr[i][j]) //converts integer value to binary
k=decimalToBinary (z[i][j])
a = [int(x) for x in str(p)]
b = [int(x) for x in str(key)]
for x=0 to 7
string = string + str(a[x] ^ b[x])
end
e_arr[i][j] = int (string [::-1], 2);
end
end
E = e_arr;
return E;
end
```

3 Results and Discussion

The efficiency of the proposed algorithm is evaluated by experimenting with simulation results and measuring against security parameters. The proposed scheme is implemented using the Python programming language and a personal computer with an Intel(R) Core (TM) i5-9400 CPU @ 2.90GHz and 8 GB RAM. The scheme is applied to the standard test images from the literature and the grayscale images from the usc-sipi dataset. The performed NCPR and UACI analysis prove that the proposed algorithm is secure. The results obtained for NPCR and UACI were compared with four recent studies taken from the literature and found that for NPCR, the proposed scheme could get the value for the minimum threshold for all input images from the dataset. In the case of UACI, the mean value of 33.5584, as shown in Table 3 , outperforms the other results. The proposed scheme is applied to standard test images of size 512x512 chosen from the literature and images from the usc-sipi data set, and the obtained result is shown in Figures 4 and 5, respectively. Figure 5 shows the scrambling and encryption process simulation results for the usc-sipi data set images 5.2.09, 5.3.01, and 7.1.06.

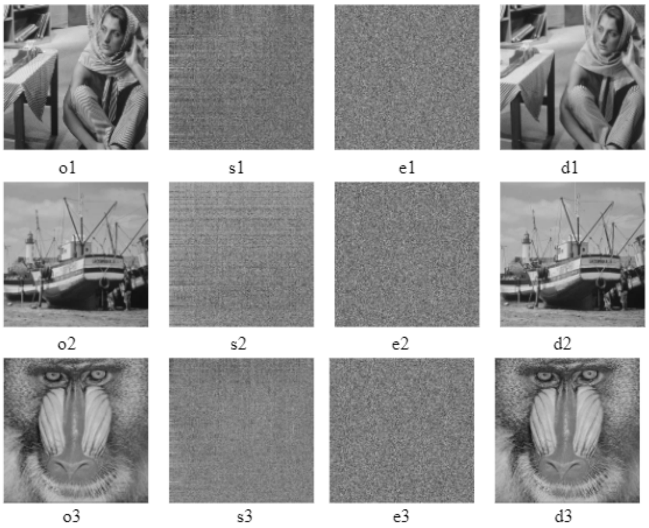


Fig 4. o1, o2, and o3 are the three standard test images, (s1,s2, s3), (e1, e2, and e3), and (d1, d2, d3) are the scrambled, encrypted, and decrypted images respectively

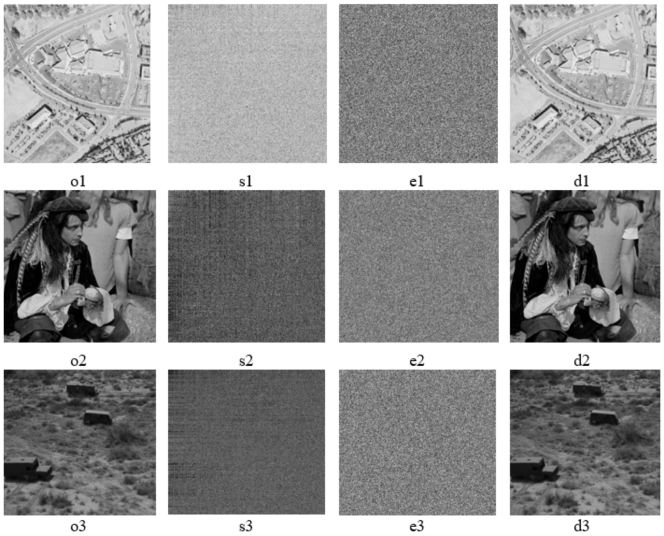


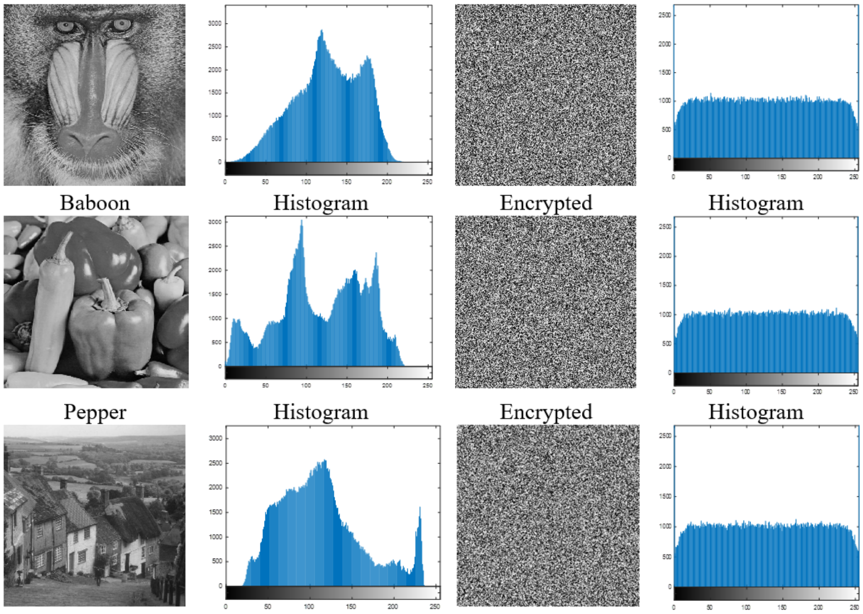
Fig 5. o1, o2, and o3 are the three images from usc-sipi data, (s1, s2, s3), (e1, e2, and e3), and (d1, d2, d3) are the scrambled, encrypted, and decrypted images respectively

3.1 Statistical Analysis

Functional MRI (fMRI) is the most sophisticated use of image statistics, where time series image frames are examined to find the signal surrounding the brain activation area. In this study histogram and correlation analysis are performed.

3.1.1 Histogram Analysis

Histogram analysis reveals that the histogram of plaintext images is uneven and prone to attacks. The histogram of an encrypted image must be uniformly distributed such that it can resist attacks to recover information by way of statistical analysis. Figure 6 shows the histogram of standard test images from the literature alongside the histogram of encrypted images. The graph shows that it is less sensitive to the attack due to the uniform distributive property. The evaluation results show that the proposed system has adequate resistance to statistical attacks, as the histogram analysis proves. The histogram of ciphertext is uniformly distributed and bears no resemblance to the plaintext histogram. The histogram analysis is enhanced with the images from the usc-sipi data set. The obtained results, as shown in Figure 7, prove the efficacy of the proposed scheme. The original and encrypted image histogram of the images 5.2.09, 5.3.01, and 7.1.06 are shown in Figure 7 as h1, h2, h3, and eh1, eh2, eh3, respectively.



Baboon Histogram Encrypted Histogram

Pepper Histogram Encrypted Histogram

Golden Hill Histogram Encrypted Histogram

Fig 6. Comparison of three standard test image histograms with the histogram of the encrypted image after applying the proposed scheme

3.1.2 Correlation Coefficient Analysis

It is ideal for image encryption to reduce correlations between nearby pixels. Through pixel scrambling or shuffling, permutation algorithms are crucial in this aspect for removing the relation between pixels. In the cipher image, a correlation coefficient value near 0 denotes a weak association between adjacent pixels, whereas a value near 1 or -1 denotes a significant correlation. Table 1 makes it evident that the technique was successful in reducing the correlation between pixels. Equation (1) calculates the correlation between neighboring pixels.

$$\gamma_{xy} = \frac{COV(X,Y)}{\sqrt{D(X)D(Y)}} \tag{1}$$

$$COV(X,Y) = \frac{1}{N}\sum_{i=1}^{N}(X_i - E(X))(Y_i - E(Y))$$

$$D(X) = \frac{1}{N}\sum_{i=1}^{N}(X_i - E(X))^2$$

$$E(X) = \frac{1}{N}\sum_{i=1}^{N}X_i$$

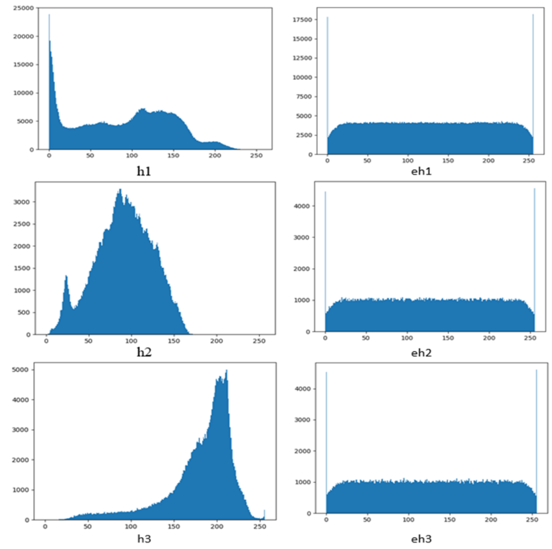


Fig 7. Comparison of the histogram of original and encrypted images of the usc-sipi data set

3.2 Differential attack analysis

A decent encryption algorithm should ensure that the associated ciphertext is entirely different when the information in the plaintext marginally changes. When the plaintext data varies slightly, a differential attack targets the method by examining the connections between different plaintexts and their related cipher texts. Two crucial metrics to assess the algorithm's resistance to differential assault are the number of pixels changes rate (NPCR) and unified average changing intensity (UACI).

Table 1. The usc-sipi data set with proposed scheme values obtained for UACI, NPCR and Correlation

Sl. No.	Images	Size	UACI (%)	NPCR (%)	Correlation
1	5.1.09	256 x 256	33.6663	99.6139	0.0126
2	5.1.10	256 x 256	33.6472	99.5437	0.0053
3	5.1.11	256 x 256	33.4926	99.3453	-0.0014
4	5.1.12	256 x 256	33.6267	99.5742	0.0132
5	5.2.08	512 x 512	33.4393	99.5296	0.0034
6	5.2.09	512 x 512	33.5586	99.5563	0.0002
7	5.2.10	512 x 512	33.5247	99.5723	-0.0014
8	7.1.01	512 x 512	33.5275	99.5548	-0.0010
9	7.1.02	512 x 512	33.4236	99.5472	-0.0009
10	7.1.03	512 x 512	33.6116	99.5468	0.0002
11	7.1.04	512 x 512	33.3607	99.4941	-0.0017
12	7.1.05	512 x 512	33.6497	99.5880	0.0015
13	7.1.06	512 x 512	33.6013	99.5624	-0.0023
14	7.1.07	512 x 512	33.6346	99.5544	-0.0005
15	7.1.08	512 x 512	33.4766	99.5571	-0.0003
16	7.1.09	512 x 512	33.6102	99.5910	0.0007
17	7.1.10	512 x 512	33.5581	99.5326	-0.0009
18	boat.512	512 x 512	33.5995	99.5571	0.0016
19	5.3.02	1024 x 1024	33.6005	99.5577	-0.0008

3.2.1 Number of Pixels Change Rate (NPCR

NPCR is used to determine the encryption algorithm's security. Considering C1 and C2 as the two images with an N × M size, we defined an array, D, with sizes similar to images C1 and C2 as mentioned in equation (2).

$$D(i,j)=\begin{cases} 0 \text{ if } C1(i,j)=C2(i,j) \\ 1 \text{ if } C1(i,j)\neq C2(i,j) \end{cases} \tag{2}$$

The NPCR determines the percentage of pixels within two different images, and it can be calculated using equation (3).

$$NPCR=\frac{\sum_{i,j} D(i,j)}{N\times M}\times 100\% \tag{3}$$

3.2.2 Unified Average Changing Intensity (UACI)

UACI determines the average intensity of the difference between the two encrypted images (C1 and C2) and it is calculated using (4).

$$UACI=\frac{1}{N\times M}\left[\sum_{i=1,j=1}^{N\times M}\frac{C_1\,(i,j)-C_2\,(i,j)}{255}\right]\times 100 \tag{4}$$

The differential analysis shows that the NPCR and UACI values are greater than 99 and 33, respectively, and are acceptable to conclude that the system is also resistant to differential attacks. The proposed scheme is tested using images from the usc-sipi data set, and the results for the NPCR, UACI, and correlation are shown in Table 1.

Table 2. Comparison of NPCR results with recent four studies

Images	**Size**	[12]	[14]	[16]	[9]	**PM**
5.1.09	256 x 256	99.6030	99.6094	99.6018	99.6109	99.6139
5.1.10	256 x 256	99.6360	99.6155	99.6245	99.5972	99.5437
5.1.11	256 x 256	99.9420	99.6094	99.6048	99.5865	99.3453
5.1.12	256 x 256	99.7920	99.5758	99.5836	99.6323	99.5742
5.2.08	512 x 512	99.9600	99.6151	99.6158	99.6105	99.5296
5.2.09	512 x 512	99.8760	99.6094	99.6405	99.6033	99.5563
5.2.10	512 x 512	99.6540	99.6166	99.5956	99.6101	99.5723
7.1.01	512 x 512	99.9570	99.5872	99.5998	99.6136	99.5548
7.1.02	512 x 512	99.9180	99.6109	99.6177	99.604	99.5472
7.1.03	512 x 512	99.8490	99.6147	99.5998	99.6101	99.5468
7.1.04	512 x 512	99.9910	99.6174	99.6108	99.6178	99.4941
7.1.05	512 x 512	99.9420	99.6212	99.601	99.5979	99.5880
7.1.06	512 x 512	99.6700	99.6475	99.6222	99.6269	99.5624
7.1.07	512 x 512	99.9830	99.6101	99.6052	99.6193	99.5544
7.1.08	512 x 512	99.8180	99.6063	99.6112	99.5979	99.5571
7.1.09	512 x 512	99.8740	99.6105	99.5968	99.6113	99.5910
7.1.10	512 x 512	99.6970	99.5907	99.5953	99.6166	99.5326
boat.512	512 x 512	99.7150	99.5998	99.6264	99.6227	99.5571
5.3.02	1024 x 1024	99.9820	99.6117	99.6078	99.6015	99.5577
Pass Count		19/19	19/19	19/19	19/19	**19/19**

A comparison of NPCR and UACI values with the four most recent studies is shown in Tables 2 and 3, respectively. The images consist of four 256 × 256 in size, 14 512 × 512 in dimension, and 1 of 1024 × 1024 in size. An unencrypted image has quite a lot of information that is susceptible to malicious attacks. Encrypted images are safer because they hide sensitive information by distorting the image in meaningful ways. Proper decryption methods are required to obtain the original image.

This unique image encryption technique is oriented toward block scrambling and logistic maps. Compared to existing chaos-based algorithms, the algorithm presented in this study has the following advantages: The block scrambling algorithm is applied

Table 3. Comparison of UACI results with recent four studies

Images	**Size**	[12]	[14]	[16]	[9]	**PM**
5.1.09	256 x 256	33.5520	33.5253	33.4446	33.4475	33.6663
5.1.10	256 x 256	33.4530	33.5115	33.4326	33.4846	33.6472
5.1.11	256 x 256	33.5860	33.5174	33.4779	33.4482	33.4926
5.1.12	256 x 256	33.4530	33.4202	33.5337	33.4453	33.6267
5.2.08	512 x 512	33.6920	33.4766	33.4336	33.5035	33.4393
5.2.09	512 x 512	33.5480	33.4528	33.4775	33.4674	33.5586
5.2.10	512 x 512	33.4540	33.3925	33.4392	33.4253	33.5247
7.1.01	512 x 512	33.6480	33.5017	33.4838	33.4885	33.5275
7.1.02	512 x 512	33.4650	33.4415	33.5015	33.4508	33.4236
7.1.03	512 x 512	33.2730	33.4455	33.4862	33.4352	33.6116
7.1.04	512 x 512	33.2020	33.4772	33.4729	33.5024	33.3607
7.1.05	512 x 512	33.8300	33.4615	33.5005	33.4739	33.6497
7.1.06	512 x 512	33.6270	33.5036	33.4355	33.4764	33.6013
7.1.07	512 x 512	33.6090	33.3952	33.5229	33.431	33.6346
7.1.08	512 x 512	33.3750	33.4682	33.4491	33.4997	33.4766
7.1.09	512 x 512	33.5300	33.394	33.5308	33.463	33.6102
7.1.10	512 x 512	33.4380	33.5565	33.4018	33.4701	33.5581
boat.512	512 x 512	33.3740	33.4519	33.4449	33.4448	33.5995
5.3.02	1024 x 1024	33.5140	33.4393	33.4877	33.4255	33.6005
Pass Count		17/19	19/19	19/19	19/19	**19/19**
Mean		33.50648	33.4648	33.4714	33.4622	**33.5584**

to scramble and cipher the pixel values. The Block-XOR scrambling strengthens the chaos-based encryption. The result shows that the image quality was not affected even after the image was decrypted. The result analyses and comparisons with the four most modern encryption methods demonstrate the strength of the suggested system in safeguarding the attributes of images.

Image encryption using pixel scrambling with chaos based on CSBSST and CDSD applied on grayscale images with subblock scrambling introduced based on a spiral transformation[10]. The study by[4,18] utilized a zig-zag model for the confusion. The image is divided into sub-images in the binary form, and a random pixel image shuffle is done. A finite state machine model is proposed for the chaotic process in[12]. The study experienced twenty-eight images from the data set and found that the finite state machine model could perform above the expected result for twenty images. Another study makes a two-dimensional chaotic map with two rounds of permutation and diffusion for encryption[14]. The average UACI obtained with the dataset images is 33.46, whereas the study proposed in this scheme of block scrambling based-on logistic map encryption could achieve a mean value of 33.55. Moreover the experiment by[16], the average UACI value is 33.45. The review in[19] discusses different chaos encryption algorithms with their performance concerning the NPCR and UACI values. The pass count values obtained for NPCR and UACI metric as shown in Table 2 and 3 prove the strength of the encryption process against different attacks.

4 Conclusion

In this study, a chaotic encryption system combined with two-pass Block-XOR operations followed by scrambling techniques is proposed to encrypt grayscale images, thus ensuring their security against malicious attacks. Plaintext images contain large amounts of sensitive information, leading to severe consequences when accessed by unintended users. Image encryption conceals this information by using encryption techniques to distort the image and reduce its resemblance to the original image. The obtained NPCR value above 99 and UACI value above 33 proves that the encrypted image is safe for transmission, such as over a digital network, and can withstand cyber-attacks. The proposed method combines block-based scrambling with chaotic-map-based encryption to generate a cipher image. The experimental results show that the proposed system provides acceptable values for all measurements.

References

1) Chaudhary N, Shahi TB, Neupane A. Secure Image Encryption Using Chaotic, Hybrid Chaotic and Block Cipher Approach. *Journal of Imaging*;8(6):167. Available from: https://doi.org/10.3390/jimaging8060167.
2) Li R, Liu Q, Liu L. Novel image encryption algorithm based on improved logistic map. *IET Image Processing*. 2019;13(1):125–134. Available from: https://doi.org/10.1049/iet-ipr.2018.5900.
3) Pourasad Y, Ranjbarzadeh R, Mardani A. A New Algorithm for Digital Image Encryption Based on Chaos Theory. *Entropy*. 2021;23(3):341. Available from: https://doi.org/10.3390/e23030341.
4) Ramasamy P, Ranganathan V, Kadry S, Damaševičius R, Blažauskas T. An Image Encryption Scheme Based on Block Scrambling, Modified Zigzag Transformation and Key Generation Using Enhanced Logistic—Tent Map. *Entropy*. 2019;21(7):656. Available from: https://doi.org/10.3390/e21070656.
5) Arshad A, Id SS, Ali A, Eleyan A. Chaos theory and its application: An essential framework for image encryption. *Chaos Theory and application*. 2020;2(1):17–22. Available from: https://www.researchgate.net/publication/341057337_Chaos_Theory_and_its_Application_An_Essential_Framework_for_Image_Encryption.
6) Ye G, Wu H, Jiao K, Mei D. Asymmetric image encryption scheme based on the Quantum logistic map and cyclic modulo diffusion. *Mathematical Biosciences and Engineering*. 2021;18(5):5427–5448. Available from: https://doi.org/10.3934/mbe.2021275.
7) Ali TS, Ali RS. A novel color image encryption scheme based on a new dynamic compound chaotic map and S-box. *Multimedia Tools and Applications*. 2022;81(15):20585–20609. Available from: https://doi.org/10.1007/s11042-022-12268-6.
8) Arif J, Khan MA, Ghaleb B, Ahmad J, Munir A, Rashid U, et al. A Novel Chaotic Permutation-Substitution Image Encryption Scheme Based on Logistic Map and Random Substitution. *IEEE Access*. 2022;10:12966–12982. Available from: https://doi.org/10.1109/access.2022.3146792.
9) Wang X, Guan N, Yang J. Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map. *Chaos, Solitons & Fractals*. 2021;150:111117. Available from: https://doi.org/10.1016/j.chaos.2021.111117.
10) Xian Y, Wang X, Yan X, Li Q, Wang X. Image Encryption Based on Chaotic Sub-Block Scrambling and Chaotic Digit Selection Diffusion. *Optics and Lasers in Engineering*. 2020;134:106202. Available from: https://doi.org/10.1016/j.optlaseng.2020.106202.
11) Broumandnia A. Designing digital image encryption using 2D and 3D reversible modular chaotic maps. *Journal of Information Security and Applications*. 2019;47:188–198. Available from: https://doi.org/10.1016/j.jisa.2019.05.004.
12) Alawida M, Teh JS, Samsudin A, Alshoura WH. An image encryption scheme based on hybridizing digital chaos and finite state machine. *Signal Processing*. 2019;164:249–266. Available from: https://doi.org/10.1016/j.sigpro.2019.06.013.
13) Kamal ST, Hosny KM, Elgindy TM, Darwish MM, Fouda MM. A New Image Encryption Algorithm for Grey and Color Medical Images. *IEEE Access*. 2021;9:37855–37865. Available from: https://doi.org/10.1109/ACCESS.2021.3063237.
14) Zhu H, Zhao Y, Song Y. 2D Logistic-Modulated-Sine-Coupling-Logistic Chaotic Map for Image Encryption. *IEEE Access*. 2019;7:14081–14098. Available from: https://doi.org/10.1109/ACCESS.2019.2893538.
15) Wang X, Li Y, Jin J. A new one-dimensional chaotic system with applications in image encryption. *Chaos, Solitons & Fractals*. 2020;139:110102. Available from: https://doi.org/10.1016/j.chaos.2020.110102.
16) Xian Y, Wang X. Fractal sorting matrix and its application on chaotic image encryption. *Information Sciences*. 2021;547:1154–1169. Available from: https://doi.org/10.1016/j.ins.2020.09.055.
17) Xiang H, Liu L. An improved digital logistic map and its application in image encryption. *Multimedia Tools and Applications*. 2020;79(41-42):30329–30355. Available from: https://doi.org/10.1007/s11042-020-09595-x.
18) Wang X, Guan N. A novel chaotic image encryption algorithm based on extended Zigzag confusion and RNA operation. *Optics & Laser Technology*. 2020;131:106366. Available from: https://doi.org/10.1016/j.optlastec.2020.106366.
19) Patro KAK, Acharya B, Nath V. Secure, Lossless, and Noise-resistive Image Encryption using Chaos, Hyper-chaos, and DNA Sequence Operation. *IETE Technical Review*. 2020;37(3):223–245. Available from: https://doi.org/10.1080/02564602.2019.1595751.