Journal of Cyber Security
DOI: 10.32604/jcs.2022.034630
Article

An Efficient Image Cipher Based on 2D Scrambled Image and Random Numbers

Asghar Ali[1,*], Sammia Ansar[1], Ashwag Albakri[2], Nadeem Iqbal[3] and Shahid Yousaf[3]

[1]Department of Mathematics, Mirpur University of Science and Technology, Mirpur, 10250, Pakistan
[2]Department of Computer Science, Jazan University, Jazan, 45142, Saudi Arabia
[3]Department of Computer Science & IT, The University of Lahore, Lahore, 54000, Pakistan
*Corresponding Author: Asghar Ali. Email: drali@must.edu.pk
Received: 01 September 2022; Accepted: 02 September 2022

Abstract: Security of images plays an import role in communication in current era due to the popularity and high usage of multimedia content in the Internet. Image security is described as applying an encryption algorithm over the given plaintext images to produce cipher images that can be transmitted safely over the open channel, the Internet. The problem which plagues these image ciphers is that they are too much time consuming, and that do not meet the dictates of the present times. In this paper, we aim to provide an efficient image cipher. The previous studies employed many constructs like Langton's Ant, 15 puzzle game and Castle in the 2D scrambled image based image ciphers, which had grave implications related to the high execution time of the ciphers. The current study directly made use of the 2D scrambled image to realize the purpose. Moreover, no compromise has been made over the security of the proposed image cipher. Random numbers have been generated by triggering the Intertwining Logistic Chaotic map. The cipher has been subjected to many important validation metrics like key space, information entropy, correlation coefficient, crop attack and lastly time complexity to demonstrate its immunity to the various attacks, and its real world application. In this paper, our proposed image cipher exhibits an encryption speed of 0.1797 s, which is far better than many of the existing encryption ciphers.

Keywords: Encryption; cipher; random numbers; image processing; secret key; chaotic map

1 Introduction

Information and Communication Technologies are changing the way we travel, entertain, live, communicate, and carry out research etc. In all those activities, images have attained an important status. Those images are being generated all over the world. We routinely share the images to our friends, colleagues, relatives and other persons. Besides, those images are also stored on diverse electro-mechanical gadgets. Sometimes, these images have a great value and importance due to their strategic dimension, the image of some spy, the image of newly developed missile for instance. So saving these

images and their transmission through some open network like the Internet is fraught with danger and threat from the community of adversaries and hackers. Hence, appropriate measures need to be taken to safeguard these images from the opponents who always seek an opportunity to have an unauthorized access over them. Classically, the niche of encryption has served this purpose like DES, AES, RSA etc. But the problem with these ciphers is that they can only encrypt textual data [1]. Whereas, the images enjoy drastically distinct properties like large volume, robust inter-pixel bonding, redundancy etc. [2]. So, it requires an entirely different framework. Fortunately, chaotic maps and their systems can spawn excellent random data. This random data is used in conducting the two necessary operations over the pixels of the given image, i.e., confusion and diffusion. There are various kinds of chaotic systems such as: low-dimensional [3], high-dimensional [4]. The systems consisting of at most two streams are dubbed as low-dimensional systems, whereas, the ones having more than two dimensions are called the high-dimension chaotic systems.

Many attacks have been launched over the image ciphers. For instance, a cipher presented in [5] was written based on only a operation of permutation. Thus, the cipher cannot avert the chosen and known plaintext attacks. The reason is that they permute the various positions of the pixels of the given image and do not make any changes in their intensity values. Hence, it is required that the new ciphers must be furnished with both the permutation and substitution operations. Moreover, other lacunas also exist in the image encryption algorithms written in previous years. As an example, by igniting the chosen plaintext attack, chosen cipher text attack and differential attack, the schemes [6–8] were cracked by [7,9,10] respectively. The primary cause of their breakage was that the streams of random numbers being employed were unrelated to the plain image which was being encryption in this scheme. To put in other words, the necessary feature of plaintext sensitivity was overlooked. Hence, the plaintext sensitivity feature must be embedded in the future ciphers. Many image ciphers have been written based on the scrambled images [1,11–13]. In [11], Langton's Ant cellular automaton has been employed in combination of the 2D scrambled image to develop an image cipher. Although the security features of this cipher are encouraging but the computational time is not much efficient due to the intermediate construct of Langton's Ant cellular automaton. In the same way, the ciphers given in [1,12] use the 2D (with 15 puzzle AI game) and 3D scrambled images to develop image ciphers which again take much computational time. Based on the novelty of these works [1,11,12], the current work also uses the 2D scrambled image to develop an yet another image cipher without using any intermediate construct like Langton's Ant cellular automaton and 15 puzzle AI game. Similarly, the work given in [13] uses chess piece castle to encrypt the image. Our goal is to propose an image cipher that can produce encrypted images faster that the existing image ciphers to get aligned with the needs of the present times.

Computational time is a very important metric that we need to consider when developing image ciphers. The ciphers that take less amount of time for their execution are more suitable for the real world applications. Several image ciphers take too much time [14] which is not fast enough to meet the requirements of the applications in the current era. Thus, less execution time is required when designing image ciphers without compromising the necessary security features.

The rest of the paper is organized as follow: Section 2 describes the fundamentals of the current study. The fundamentals are: the chaotic systems, which provide the streams of random numbers. A suggested image encryption algorithm and its corresponding decryption algorithm have been explained in Section 3. The simulation and computer experiments are presented in Section 4. Section 5 exhibits the performance and the security analysis of the proposed cipher given in Section 3 to demonstrate its do-ability. Lastly, conclusion and future directions are depicted in Section 6.

2 Fundamentals

This section discusses the fundamentals upon which the current study rests. In particular, we describe the chaotic systems which are extensively being employed in the discipline of image encryption.

2.1 Chaotic Systems

Chaotic systems are the life blood of virtually tens of hundreds of encryption schemes for images written in the last two decades. Those systems provide the chaotic streams necessary to carry out the operations of scrambling and substitution. The marvelous properties of these systems are: extreme sensitivity to the initial states and the parameters of the system, pseudorandom, mixing, arbitrariness etc. To put in other words, a very faint twist in any one of the initial states and parameters of the system causes to render the absolutely different results. A lot of chaotic systems exist. The chaotic system which fits our logic is the Intertwining Logistic Map [1] described in the following equation.

$$\begin{cases} x_{n+1} = [\,\mu \times k_1 \times y_n \times (1 - x_n) + z_n]\, mod\ 1 \\ y_{n+1} = \left[\mu \times k_2 \times y_n \times z_n \times \dfrac{1}{1} + x_{n+1}^2\right] mod\ 1 \\ z_{n+1} = [\,\mu \times (x_{n+1} + y_{n+1} + k_3) \times sinz_n]\, mod\ 1 \end{cases} \tag{1}$$

The above equation provides three streams of random numbers, if these conditions hold: $0 < u <= 3.999$, $|k_1| > 33.5$, $|k_2| > 37.9$, $|k_3| > 35.7$. This map would enjoy better chaotic behavior if it is put in the context of its previous logistic map. Moreover, it does not have any blank window. Additionally, it keeps even distribution [15].

3 The Proposed Image Encryption Scheme

This work intends to provide an algorithms for the encryption and decryption of the plain images. We assume the dimensions of this images would be m × n. Fig. 1 shows the flow of the algorithm this study has conceived.

The proposed encryption algorithm works as follows: first a sum of all the input pixels of the gray scale image is computed. Then, the sum and a secret key are input to the module of *Adding the plaintext sensitivity*. Then, the tempered secret key has been given to the intertwining logistic map. This map renders three random numbers' streams, i.e., *is called u, v and w*, respectively. The first two streams u and v along with the plain input image are provided to the module called *Image Scrambler* which carries out the necessary scrambling operation of the pixels of the given image. Finally, the scrambled image and the third stream of random numbers, i.e., w, are XORed to produce the final cipher image.

This scheme works on the philosophy of 2D scrambled image D. This image D has been initially set to −1 for all its pixels. Further, the plain gray scale image has been resized to 1 × mn. Now, the pixels taken from the plain gray scale image are inserted randomly on the different locations of the scrambled image. The two streams of random numbers u and v have been used for this purpose. Lastly, the vacant cells of the scrambled image have been populated through the remaining pixels of the 1D plain image.

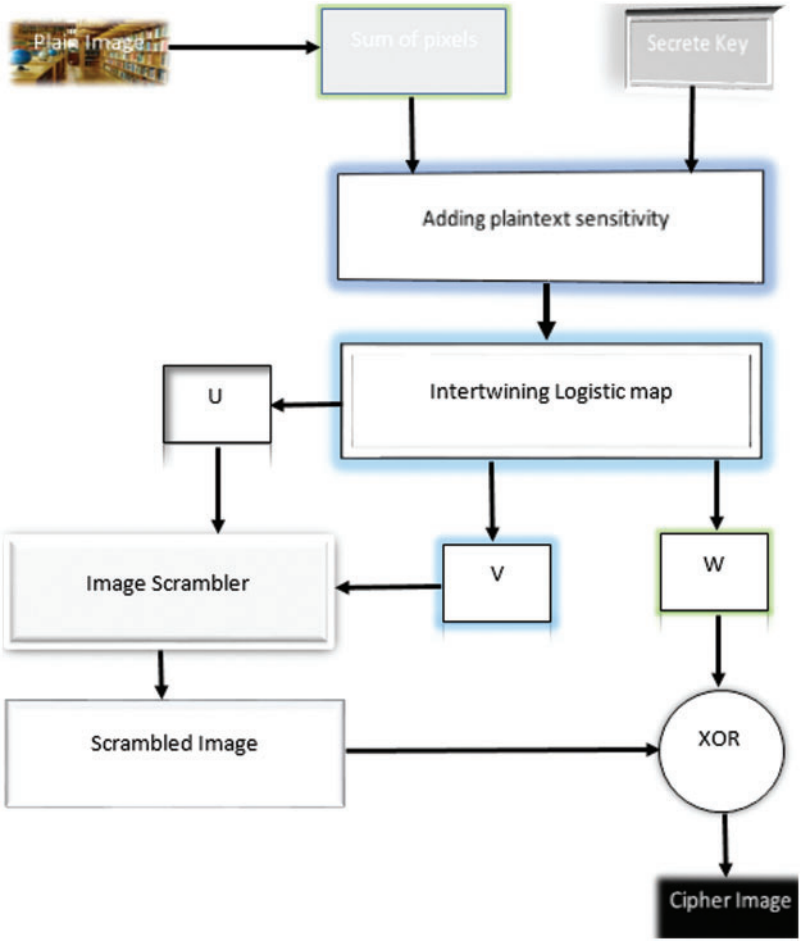

Figure 1: The proposed encryption scheme

3.1 Generation of Key Streams

Hackers usually resort to an attack called the *differential attack*. The differential attack is curbed through the introduction of plaintext sensitivity in the anatomy of the algorithm. In our proposed work, we utilize an image characteristic called sum to embed the feature of the plaintext sensitivity. The sum refers to the sum of all the pixels of the given plain image. This sum has been used as follows.

Step 1:

$$x_0 = x\prime_0 + \frac{Sum\,(Img)}{10^{10}} \tag{2}$$

where x'_0 denotes the value before adding the plaintext sensitivity and x_0 the value after adding the plaintext sensitivity.

Step 2: By sparking the chaotic maps/systems given in Eq. (1), these are the streams of chaotic data which have been obtained: $\{u\,(t)\}_{t=1}^{mn}$, $\{v\,(t)\}_{t=1}^{mn}$ *and* $\{w\,(t)\}_{t=1}^{mn}$. Here, (m, n) represents the size of the input image.

Step 3: Step 2 rendered the chaotic sequences u, v and w in a "rudimentary" form. They must be tailored before their usage to gain their utility. So the following Eq. (3) perform this duty to obtain the sequences U, V and W.

$$\begin{cases} U(i) = floor(mod\left(abs\left(u(i)\right) - floor\left(abs\left(u(i)\right)\right) \times 10^{14},\ m\right) + 1), \\ V(i) = floor(mod\left(abs\left(v(i)\right) - floor\left(abs\left(v(i)\right)\right) \times 10^{14},\ n\right) + 1), \\ W(i) = floor(mod\left(abs\left(w(i)\right) - floor\left(abs\left(w(i)\right)\right) \times 10^{14},\ 256\right)) \end{cases} \tag{3}$$

where u(i), v(i), w(i) are the corresponding elements of u, v, w respectively. mod(s, t) gives the remainder when s is divided by t. i = 1, 2, . . . , mn.

3.2 Image Encryption Scheme

Call the Algorithm 1 with the parameters IMG, U, V and W.

Algorithm 1: Image Scrambler based on the 2D Scrambled image

```
    Input: IMG, U, V, W
    Output: IMG'
1   [M, N] = size (IMG)
2   IMG = reshape(IMG, [I, M × N])
3   I = 1
4   While I <= M do
5    J = 1
6    While I <= N do
7          D(I, J) = −1
8         J = J + 1
9   I = I + 1
10   K = 1
11   While k <= MN do
12                I = U(k)
13                J = V(k)
14                if D(I, J) == −1 then
15               D(I, J) = IMG(K)
16               IMG(K) = −1
17   KK = 1
18   I = 1
19   While I <= M do
20              J = 1
21              While j <= M do
22                      if (D(I, J) == −1 then
23                          While img(k) = −1 do
24                                KK = KK + 1
25                          D(I, J) = IMG(KK)
26                          IMG(KK) = −1
27                          KK = KK + 1
```

(Continued)

Algorithm 1: Continued

```
28                    J = J + 1
29              I = I + 1
30    I = 1
31    While I <= M do
32              j = 1
33              While j <= N do
34                    IMG` (I, J) = D(I, J) ⊕ w(I, J)
35                    J = J + 1
36              I = I + 1
```

Here we give an explanation of the Algorithm 1. Line 1 finds the size of the given image IMG. Let the calculated size is (M, N). Lines (3–9) initialize the scrambled image D with −1. Lines (11–16) use the *while* loop which iterates MN times. On each K[th] iteration, it assigns the random numbers to (I, J) from the streams of random numbers U and V. Line 15 assigns the K[th] pixel to the $D(I, J)$. Further, line 16 assigns −1 to IMG(K) to keep the track. Lastly, the lines (19–29) assign the remaining pixels of the image IMG to the scrambled image D. These were the instructions of the proposed algorithm to carry out the scrambling operations. Lines (31–36) perform the diffusion effects to the scrambled image D. In particular, the XOR operation is being carried out on the line 34. Lastly, IMG' is the final cipher image.

3.3 Image Decryption Procedure

Image decryption algorithm has been shown in the Fig. 2. Since the proposed image cipher has been written based on the principles of private key cryptography, so the decryption steps are inverse of the encryption algorithm. Now, the cipher image is input to the decryption algorithm. On the other side, secret key has been given to the intertwining logistic map which renders the three streams of random numbers namely U, V and W. Call the Algorithm 2 with the parameters IMG', U, V and W. Most of the steps of this algorithm are same as of the encryption algorithm. Here, the first operation would be the nullifying the diffusion effects which is being conducted in the line 7. The resultant image is IMG.

Algorithm 2: Image Decryption

```
    Input: IMG', U, V, W
    Output: PI
1    [M, N] = size (IMG')
2    IMG = reshape(IMG', [I, M × N])
3    I = 1
4    While I <= M do
5     J = 1
6     While I <= N do
7          IMG(I, J) = IMG'(I, J) ⊕ w(I, J)
8          J = J + 1
9     I = I + 1
10   I = 1
11   While I <= MN do
12         PI(I) = −1
```

(Continued)

Algorithm 2: Continued

```
K = 1
While K <= MN do
        I = U(K)
        J = V(K)
        if IMG(I, J) ≠ −1 then
            PI(K) = IMG(I, J)
            IMG(I, J) = −1
index = 1
I = 1
While I <= M do
        J = 1
        While j <= N do
            if PI(index) ≠ −1 then
                While PI(index) ≠ −1 Do
                    index = index + 1
                PI(index) = IMG(I, J)
                IMG(I, J) = −1
                index = index + 1
            J = J + 1
        J = J + 1
PI = reshape (PI, [M, N])
```

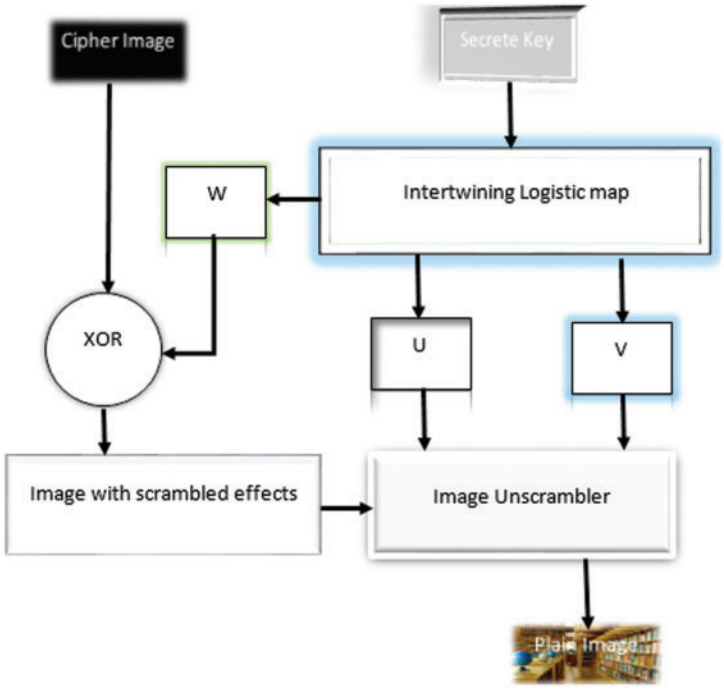

Figure 2: Decryption scheme

In Algorithm 2, line 12 initializes the plain image *PI* with the value −1 for the size of *MN*. Now the lines (14–33) shift the pixels from the image *IMG* to the plain image *PI* which outputs the plain image that are identical to the original image given to the encryption algorithm.

4 Experiments and Simulation

This section exhibits the practical utility of the theoretical framework described in the previous section. To serve this purpose, four test images namely Lena, Baboon, Brain and Butterfly are chosen from USC-SIPI images repository. MN is the size (length and width) of those images. The tool used is MATLAB 2016. Further, it is based on 64-bit double-precision. The secret key given to the proposed image cipher is $x_0 = 0.36$, $y_0 = 0.25$, $z_0 = 0.78$, $k_1 = 35.5$, $k_2 = 38.2$, $k_3 = 36.0$. Four chosen plain images are drawn in the Figs. 3a to 3d. Cipher and decrypted images can be seen in the Figs. 3e to 3h and the Figs 3i to 3l, respectively. The figures expressly exhibit the successful decryption as well as encryption algorithms for the proposed image cipher.

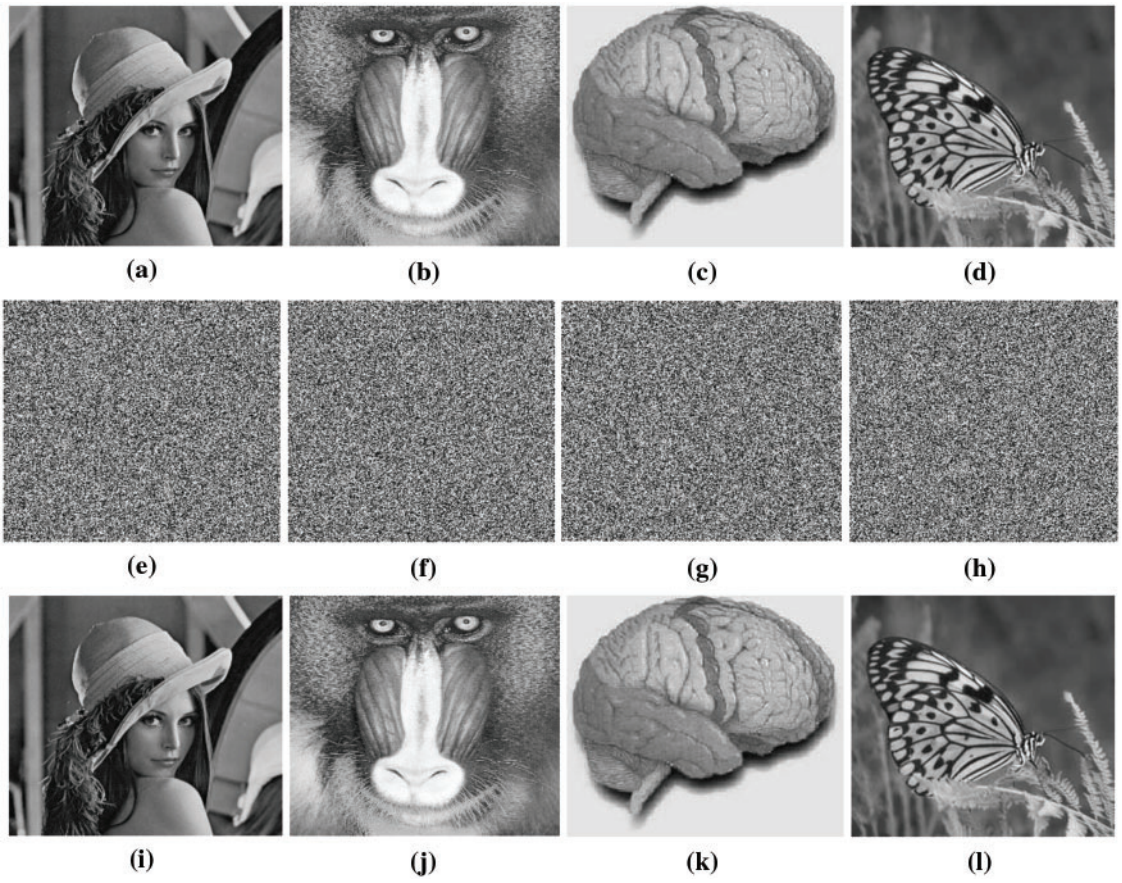

Figure 3: Plain, cipher and decrypted images: (a) Lena plain image; (b) Baboon plain image; (c) Brain plain image; (d) Butterfly plain image; (e) Lena cipher image; (f) Baboon cipher image; (g) Brain cipher image; (h) Butterfly cipher image; (i) Lena decrypted image; (j) Baboon decrypted image; (k) Brain decrypted image; (l) Butterfly decrypted image

5 Security Analysis and Validation

Just the development of some cipher is not sufficient, rather this cipher must be subjected to the different validation metrics in order to verify its authenticity. This is what we are going to carry out in this section. Apart from that, several research work [11,16–18] have been taken from the literature for the sake of comparison.

5.1 Key Space Analysis

The larger key space provide a better resistant against brute force attacks. Fact of the matter is that cryptanalysts savvy generates all the possible secret keys to run the decryption algorithm. Researchers have put forward 2^{100} [16] as the minimum threshold for the key space to eliminate this threat. The variables (x_0, y_0, z_0, k_1, k_2, k_3 and u) being used in the chaotic system make up the secret key of the cipher. If one takes the computer precision as 10^{-15}, the corresponding key space calculates to be $10^{105} \approx 2^{348}$. Obviously, the proposed cipher is equipped with a large key space since $2^{348} >> 2^{100}$. Moreover, a comparison has also been made between the previous published works and the proposed cipher based on this metric (Table 1). We can observe the proposed work is better than [17,18] based on the validation metric of the key space.

Table 1: Proposed algorithm key space and comparison with other works

Algorithms	Key space
Proposed	$10^{105} \approx 2^{348}$
Ref. [11]	10^{141}
Ref. [16]	10^{105}
Ref. [17]	2^{124}
Ref. [18]	2^{80}

5.2 Statistical Analysis

Correlation coefficient and histogram are the frequently used metrics under the statistical analysis.

5.2.1 Correlation Coefficient Analysis

Correlation coefficient (CC) refers to the *tightness* of the two consecutive/neighboring pixels of an image. Fact of the matter is that the pixels of the plain image are strongly linked with each other. So their correlation coefficient is much higher. As the cipher is applied over the pixels of an image, the intensity values of its pixels are subjected to a sweeping change in their locations as well as in their intensity values. This change is gauged through this metric. Naturally, the value of this metric comes down steeply after the cipher algorithm gets applied on the plain image. To demonstrate this metric, we have taken five thousand pixels in the diagonal, vertical and horizontal directions. Formula below is normally employed for this purpose [1].

$$CC = \frac{P\sum_{d=1}^{p}(z_d \times w_d) - \sum_{d=1}^{p} z_d \times \sum_{d=1}^{p} w_d}{\sqrt{\left(P\sum_{d=1}^{p} z_d^2 - \left(\sum_{d=1}^{p} z_d\right)^2\right)\left(P\sum_{d=1}^{p} w_d^2 - \left(\sum_{d=1}^{p} w_d\right)^2\right)}} \tag{4}$$

The intensity values of the two consecutive pixels have been denoted by z and w according to the above mathematical equation. Moreover, total number of pixels have been denoted by P. Besides, Lena's plain and cipher images pixels distribution in the vertical, horizontal and diagonal directions

have been demonstrated through the figure [4]. Apart from that, Table 2 presents this metric's results. This metric denotes the values of CC for the consecutive pixels of both cipher and plain images of Lena. According to this table, the value of CC is approximately 1 before the application of encryption algorithm over the plain image. But this value drops down and becomes nearly equal to zero. Both the Fig. 4 and Table 2 show that the potential relation between cipher and plain images dropped down with the application of cipher over the plain image. Besides, a comparative analysis has also been made (see Table 3). We can see that the stats of suggested algorithm are comparable to our selected research [11,16–18].

Table 2: Correlation coefficient results

Image	Correlation		
	Horizontal	**Vertical**	**Diagonal**
Plain image lina	0.9165	0.9521	0.9280
Encrypted image lina	0.0040	0.0033	−0.0032

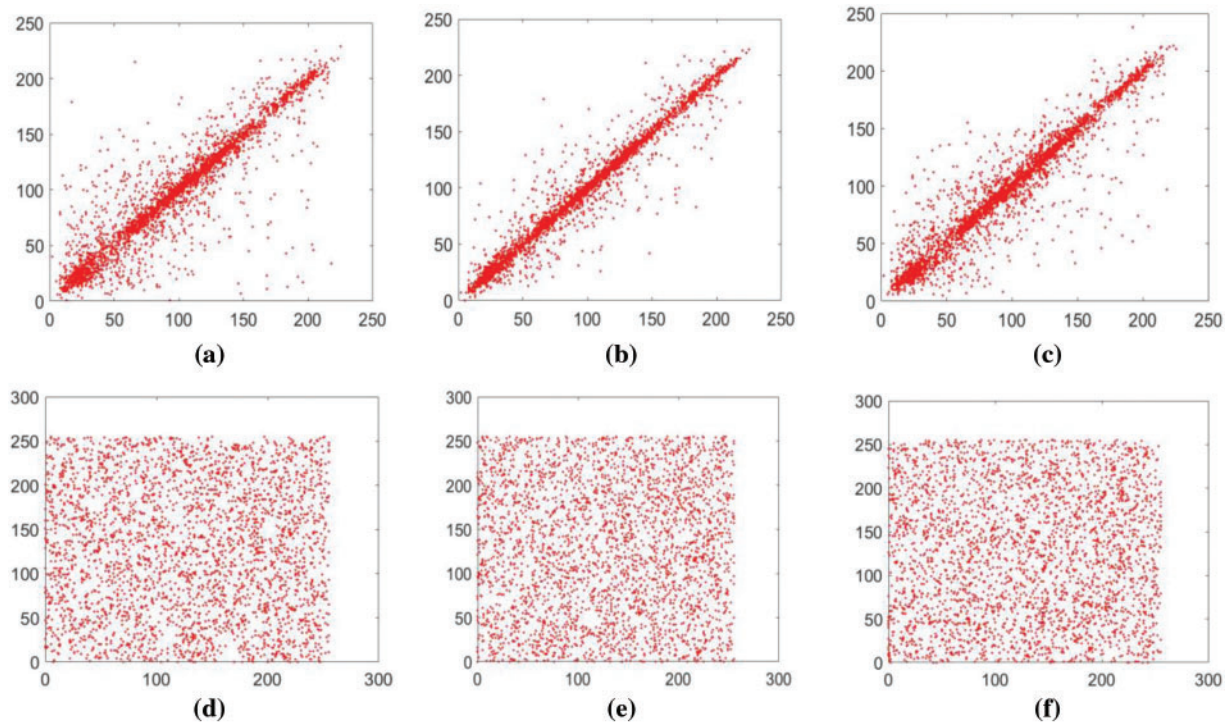

Figure 4: Lena image's correlation distribution (Direction, image type): (a) Horizontal, plain; (b) Vertical, plain; (c) Diagonal, plain; (d) Horizontal, cipher; (e) Vertical, cipher; (f) Diagonal, cipher

Table 3: Comparison of CC between the published works and the suggested algorithm

Image	Encryption scheme	Direction		
		Horizontal	**Vertical**	**Diagonal**
Plain image lena		0.9165	0.9521	0.9280
Encrypted image lena	Our algorithm	0.0040	0.0033	−0.0032
	Ref. [11]	0.00127	0.00169	−0.00154
	Ref. [16]	−0.0081	0.0035	−0.0368
	Ref. [17]	−0.005649	0.000610	0.001835
	Ref. [18]	0.0044	0.0033	0.0701

5.2.2 Histogram Analysis

The construct of histogram carries out the frequency distribution of the pixels present in the given image. Normally, histogram of a plain image has a curved and slated bar over it. This curved and slanted bar serves as a major source of information for the potential cryptanalysis savvy. He can exploit this information to materialize his nefarious designs. After a good cipher gets applied over the plain image, the pixels go through a global change in their intensity values and their relative locations. Naturally, the histogram made after it has a smooth and uniform bar over it. The characteristics of uniformity and smoothness of a bar gives a very tough time to the hackers to infer some useful information out of this histogram. Fig. 5 depicts the plain and cipher images' histograms of Lena. One can see both the slanting and smooth bar over the histograms of plain and cipher images respectively. These phenomena indicate the robustness and security of the suggested cipher. To put this in other words, this also indicates that suggested encryption algorithm is immune to the potential histogram attack.

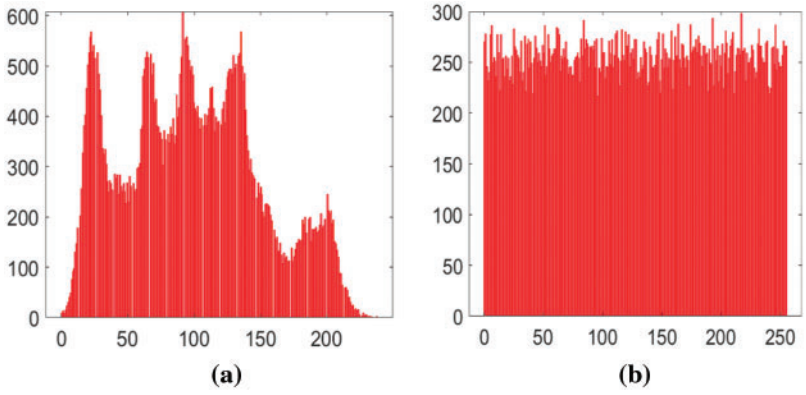

Figure 5: Histogram analysis (a) Plain lena image; (b) Encrypted lena image

5.3 Entropy Analysis

This is another validation metric to measure the extent with which the given input image has been randomized. As the given image is subjected to confusion and diffusion operations, its pixels change

both in their intensity values and locations. This dispersion is measured through that metric. Shannon, an information theorist, gave its formula in 1949 [19].

$$E(r) = \sum_{d=0}^{2^n-1} p(r_d) \log \frac{1}{p(r_d)} \tag{5}$$

According to the above equation, $E(r)$ is referring to the information entropy of the cipher image given the signal r. Apart from that, $p(r_d)$ is denoting the probability of r_d. As pixels are disturbed, its value enhances. If the pixels are disturbed to an idealistic proportions, this value comes out to be exactly 8 for the images with 256 gray values. Table 4 shows the results of this metric. Besides, Table 4 draws a comparison between the other research work and our proposed work. Although the result of the proposed work is good regarding this metric, but unfortunately, it doesn't beat any of the chosen works.

Table 4: Results for the metric information entropy and a comparison with other works

Encryption algorithm	Images	Original	Encrypted
Our algorithm	Lena	7.5835	7.9969
	Baboon	7.4769	7.9970
	Brain	6.4031	7.9971
	Butterfly	7.2353	7.9972
	Average	7.1747	7.9971
Ref. [11]	Lena	-	7.9993
Ref. [16]	Lena	7.3200	7.9973
Ref. [17]	Lena	7.5954	7.9978
Ref. [18]	Lena	7.5788	7.9973

5.4 Crop Attack Analysis

Different fears, uncertainties and other dangers characterize the general tone and tenor of our existence in this world. As the cipher image is sent to some destination or is stored on some electronic gadget, there are chances that they may undergo the crop attack. In this attack dynamics, some portion of an image is lost. Now the decryption algorithm should be much powerful to retrieve the image in such a way that the original image must be recognized. If this is the case, then the cipher under question will be saved from the potential crop attacks. To demonstrate this property, we have cut some portions of the cipher images (Figs. 6a and 6b). These images, then have been subjected to the decryption algorithm and the resultant images are drawn in the Figs. 6c and 6d. Original plain images can be easily discerned according to these figures. Hence, we assert that the suggested image encryption algorithm is resistant to future crop attacks.

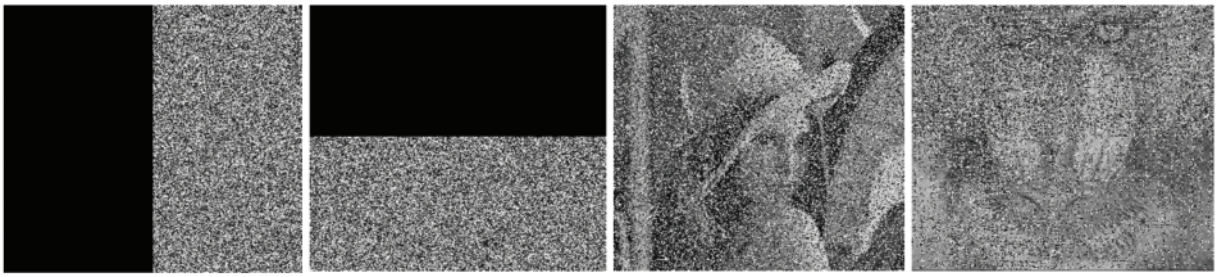

Figure 6: Crop attack analysis on cipher images: (a) Lena cipher image with data loss of 0.5 × m × n; (b) Baboon cipher image with data loss of 0.5 × m × n; (c) Decrypted image from (a); (d) Decrypted image from (b)

5.5 Speed Analysis

The specification of the platform selected for this project are: Intel(R) Core(TM) i7-3740QM CPU @ 2.70 GHz, RAM = 8.00 GB, System Type: 64-bit Operating System, x64-based processor. Besides, the underlying operating system is Windows 10. Further, the programming language chosen is MATLAB R2016a. The Table 5 shows the time taken by the suggested image cipher. Moreover, this time has also been compared with the other published research work. Our work performs better than the ones given in these works [16,17] Hence, the proposed work is much efficient as far as computational time is concerned.

Table 5: Running speed and a comparison with other ciphers

Algorithm	Images	Speed (sec)
Proposed	Lena	0.1723
	Baboon	0.1789
	Brain	0.1832
	Butterfly	0.1843
	Average	**0.1797**
Ref. [17]	Lena	17.63
Ref. [18]	Lena	0.382
Ref. [19]	Lena	0.01012

6 Conclusion

Some image ciphers have been written based on the 2D and 3D scrambled images. Although results of the majority of the security parameters are very encouraging for these ciphers but they consume too much time due to the intermediate construct being used in them. The current study has put forward a yet another image cipher based on the 2D scrambled image without using any intermediate construct that have been used in previous studies. This idea of ours cast very positive repercussions as the validation metrics suggest. More importantly, the computational time is very efficient for the proposed image cipher without compromising the other security features like key space, information

entropy, correlation coefficient etc. We contend that the suggested image cipher bears good chances for its real world application. To make the cipher more secured suggested in this study, DNA encoding can be added. This is the possible future research direction.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Iqbal, S. Abbas, M. A. Khan, T. Alyas, A. Fatima *et al.,* "An RGB image cipher using chaotic systems, 15-puzzle problem and DNA computing," *IEEE Access*, vol. 7, pp. 174051–174071, 2019.
[2] T. Li, B. Du and X. Liang, "Image encryption algorithm based on logistic and two-dimensional lorenz," *IEEE Access*, vol. 8, pp. 13792–13805, 2020.
[3] T. X. Jun, Z. Wang, M. Zhang, Y. Liu, H. Xu *et al.,* "An image encryption algorithm based on the perturbed high-dimensional chaotic map," *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1493–1508, 2015.
[4] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high dimension chaotic system," *Optics Communications*, vol. 284, pp. 16–17, pp. 3895–3903, 2011.
[5] F. Özkaynak and A. B. Özer, "Cryptanalysis of a new image encryption algorithm based on chaos," *Optik*, vol. 127, no. 13, pp. 5190–5192, 2016.
[6] W. Zhang, K. W. Wong, H. Yu and Z. I. Zhu, "A symmetric color image encryption algorithm using the intrinsic features of bit distributions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 584–600, 2013,.
[7] G. Zhou, D. Zhang, Y. Liu, Y. Yuan and Q. Liu, "A novel image encryption algorithm based on chaos and line map," *Neurocomputing*, vol. 169, pp. 150–157, 2015.
[8] W. Zhang, H. Yu, Y. I. Zhao and Z. I. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 118, pp. 36–50, 2016.
[9] T. Hoang and H. X. Thanh, "Cryptanalysis and security improvement for a symmetric color image encryption algorithm," *Optik*, vol. 155, pp. 366–383, 2018.
[10] J. Wu, X. Liao, B. Yang, "Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 142, pp. 292–300, 2018.
[11] X. Wang, and D. Xu, "A novel image encryption scheme using chaos and langton ant cellular automaton," *Nonlinear Dynamics*, vol. 79, no. 4, pp. 2449–2456, 2015.
[12] N. Iqbal, M. Hanif, S. Abbas, M. A. Khan, and Z. U. Rehman, "Dynamic 3d scrambled image based RGB image encryption scheme using hyperchaotic system and dna encoding," *Journal of Information Security and Applications*, vol. 58, pp. 102809, 2021.
[13] N. Iqbal, R. A. Naqvi, M. Atif, M. A. Khan, M. Hanif *et al.,* "On the image encryption algorithm based on the chaotic system, dna encoding, and castle," *IEEE Access*, vol. 9, pp. 118253–118270, 2021.
[14] Z. Xiong, Y. Wu, C. Ye, X. Zhang and F. Xu, "Color image chaos encryption algorithm combining crc and nine palace map," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31035–31055, 2019.
[15] I. S. Sam, P. Devaraj and R. Bhuvaneswaran, "An intertwining chaotic maps based image encryption scheme," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 1995–2007, 2012.
[16] H. Zhu, L. Dai, Y. Liu and L. Wu, "A Three-dimensional bit-level image encryption algorithm with rubik cube method," *Mathematics and Computers in Simulation*, vol. 185, pp. 754–770, 2021.
[17] Q. Lu, C. Zhu and X. Deng, "An efficient image encryption scheme based on the lss chaotic map and single s-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.

[18] M. Gupta, V. P. Singh, K. K. Gupta and P. K. Shukla, "An efficient image encryption technique based on two-level security for internet of things," *Multimedia Tools and Applications*, vol. 1, pp. 1–21, 2022.
[19] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.