



A robust and secure immensely random GAN based image encryption mechanism

Om Dev Singh¹ · Sangeeta Dhall¹ · Anjali Malik¹ · Shailender Gupta¹

Received: 20 November 2021 / Revised: 5 August 2022 / Accepted: 19 September 2022 /
Published online: 6 October 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Protection of information (data/images) is crucial in the colossal and ever-expanding domain of digital transfer. Cryptography is one of the well-known admired solutions to preserve images' confidentiality over highly unreliable and unrestricted public media. Researchers propose numerous techniques to accomplish the ever-growing need for security. In continuation, this paper aims to develop a robust image encryption scheme that accomplishes the task of protection by employing a series of specially designed substitution box, permutation box and diffusion box by taking encryption keys as input which is consequently generated from a Generative Adversarial Network (GAN), an unsupervised deep learning algorithm trained on the Logistic Maps. The substitution box performs byte-level substitution using two different schemes, and the other two perform encryption at both bit-level and byte-level, which helps it withstand a wide range of attacks. A dataset with 789 standard images is taken for experimentation, partitioned into three sets according to size (128, 256, and 512). The projected scheme outperforms state-of-the-art methods with better performance since the trained generator passed the comprehensive tests; it also withstands most of the probable attacks available in the literature. GAN was subjected to the chi-square test, runs test, and NIST test suite to check the randomness of the Pseudo-Random Number Generator. The projected algorithm offers promising visual, statistical, robustness, and quantitative analysis results.

✉ Sangeeta Dhall
Sangeeta_dhall@yahoo.co.in

Om Dev Singh
devsingh640@gmail.com

Anjali Malik
anjalimalik0611@gmail.com

Shailender Gupta
Shailender81@gmail.com

¹ J.C. Bose University of Science and Technology, YMCA, Faridabad, India

Keywords Image encryption · Generative adversarial network (GAN) · Unsupervised learning · Deep learning · Logistic maps · Pseudo random number generator (PRNG)

1 Introduction

The introduction of the internet in the early 1990s by the United States equipped users with the ability to communicate and share data at no matter of time. However, due to a lack of knowledge and availability of suitable protection mechanisms, this freedom results in the interception of intruders in accessing digital data. The scenario resulted in the need for a suitable security mechanism that must fulfil three primary goals; Confidentiality, Integrity, and Availability; lack of any of the entities from the triad may lead to data loss or, in the worst case, provide access to the complete system to a hacker or cracker. It is imperative to design a security mechanism to resist both the active and passive attacks employed by the attacker to gain access. In terms of image encryption schemes, this can be achieved by employing high brute force search time, large key-space, high key sensitive mechanism, and a highly randomized pseudo-random number generator (PRNG), which altogether makes it very difficult for a hacker or cracker to penetrate through.

This paper aims to develop an impenetrable image encryption scheme to provide the solution to the problems discussed. The proposed scheme offers a very high brute force search time, a very large key-space, a very high key sensitive mechanism, and multi-level encryption, and the scheme also employs a highly randomized random number generator which is an unsupervised deep learning algorithm trained on random sequences generated by traditional logistic maps. To achieve this, the proposed image encryption scheme consists of highly randomized key-dependent confusion and diffusion processes; both processes use highly randomized sets of keys generated by the GAN [7, 9, 13, 17, 24, 35] at the individual level, making it impossible to find the way back to the valuable input image.

The rest of the paper is discussed as follows: **Section 2, “Literature Survey,”** scatters light on the past and present developments in the field of image encryption based on image processing and deep learning; it also focuses on the adoption of various evaluation matrices for better understanding and comparison. **Section 3, “Proposed Image Encryption Scheme,”** provides a brief discussion on the developed image encryption scheme. **Section 3.4, “Evaluation Metric,”** revolves around pre-setting a metric to justify the robustness of an image encryption technique based on image processing and deep learning. **Section 4, “Results,”** presents the result for the proposed technique based on the pre-set Evaluation Metric and a brief comparison with the techniques present in the literature. **Section 5.6.1, “Conclusion,”** presents the overall conclusion deduced during the research and the comparison done. **Section 5.6.2, “Motivation,”** contains the papers that this research is inspired with, followed by the list of **References**.

Table 1 represents the list of abbreviations that are used throughout this paper.

2 Literature survey

This section discusses the numerous image encryption schemes based on image processing and deep learning [28–30]. The overall trends of the research in the encryption field are listed in Table 2. In I. Shatheesh Sam et al.in [40], the proposed scheme employed intertwining chaotic

Table 1 Abbreviation table

Serial Number	Acronym	Full Form
1	GAN	Generative Adversarial Network
2	PRNG	Pseudo Random Number Generator
3	NIST	National Institute of Standards and Technology
4	RGB	Red Green Blue
5	LSB	Least Significant Bit
6	XOR	Exclusive OR
7	VA	Visual Analysis
9	SA	Statistical Analysis
10	QA	Quantitative Analysis
11	DAA	Differential Attack Analysis
12	OAA	Other Attack Analysis
13	RTA	Randomness Test analysis
14	IP	Image Processing
15	VI	Visual Inspection
16	CC	Correlation Coefficient
17	HA	Histogram Analysis
18	H	Entropy
19	MAE	Mean Absolute Error
20	NPCR	Number of Pixels Change Rate
21	UACI	Unified Average Changing Intensity
22	BFST	Brute Force Search Time Attack
23	KSA	Key Sensitivity Analysis
24	TestU01	TestU01: A C Library for Empirical Testing of Random Number Generators
25	PSNR	Peak Signal to Noise Ratio
26	MSE	Mean Square Error
27	VAR	Variance
28	NA	Noise Attack
29	AOA	Anti-Occlusion Attack
30	CA	Crypto Analysis
31	BER	Bit Error Rate
32	GA	Geometrical Attack
33	CST	Chi-Square Test
34	JCA	Jpeg Compression Attack
35	CNN	Convolution Neural Network
36	SSIM	Structural Similarity Index
37	COA	Cipher Only Attack
38	CPA	Chosen Plaintext Attack
39	KPA	Known Plaintext Attack
40	EHCC	Encrypted Horizontal Correlation Coefficient
41	EVCC	Encrypted Vertical Correlation Coefficient
42	EDCC	Encrypted Diagonal Correlation Coefficient
43	OHCC	Original Horizontal Correlation Coefficient
44	OVCC	Original Vertical Correlation Coefficient
45	ODCC	Original Diagonal Correlation Coefficient
46	EOI	Entropy of Output Image
47	E EI	Entropy of Encrypted Image
48	DL	Deep Learning
49	HBH	Horizontal Bottom Half
50	HUH	Horizontal Upper Half
51	VLH	Vertical Lower Half
52	VRH	Vertical Right Half
53	BLT	Bottom Lower Triangle
54	URT	Upper Right Triangle
55	BRT	Bottom Right Triangle
56	ULT	Upper Left Triangle
57	OII	Original Input Image

Table 1 (continued)

Serial Number	Acronym	Full Form
58	VFI	Vertically Flipped Image
59	HFI	Horizontally Flipped Image
60	VHFI	Vertically and Horizontal Flipped Image
61	CWR-90°	Clock Wise 90° Rotation
62	CWR-180°	Clock Wise 180° Rotation
63	ACWR-90°	Anti-Clock Wise 90° Rotation

maps, in which the confusion process consists of permutation and byte-level substitution followed by a diffusion box that performs nonlinear diffusion and sub-diagonal diffusion.

In X. Liao et al. [25] the algorithm employs wave's superposition principle and *XOR* operation. The input original image is split into two equal halves; and each half is encrypted using wave transmission encryption with four waves. A. Akhshani et al. in [3] proposed an encryption based on the quantum logistic maps. The scheme makes use of the dissipative quantum systems for the development of image encryption mechanism based on quantum logistic maps. I. Sam et al. in [39] proposed a scheme for encryption of coloured images and can support key sizes between 192 to 400 *bits*. The technique is based on the transformed logistic maps and consists of confusion and diffusion processes. In confusion box the permutation of image pixels takes place based on the six odd keys and further permuted using the first chaotic key in nonlinear diffusion box followed by the *XORing* of nonlinear diffusion box output with the second chaotic key, finally the third key is used in the zig-zag diffusion process. M. François et al. in [16] proposed an image encryption scheme based on coupling of chaotic function and *XOR* function. The algorithm employs chaotic function that uses linear congruence's. A. A. Abd El-Latif et al. in [2] proposed a quantum chaotic system based colour image encryption scheme. The scheme composed of a substitution box and a diffusion box, in former operation is done on the basis of toral automorphism in integer wavelet transform followed by a diffusion box developed by mixing the feature of horizontally and vertically adjacent pixels on the basis of quantum chaotic map. At last, substitution is accomplished by creating an intermediate chaotic keystream image with quantum chaotic system. M. SaberiKamarposhti et al. in [38] proposed an image encryption scheme based on biological operations and uses a 120 - *bit* secret key. Initially the image pixels were scrambled using the Deoxyribonucleic Acid (DNA) sequences and the cyclic chaos followed by the pixel value modification based on a mask generated by the cyclic chaos. In R. Bansal et al. [8] the proposed scheme grounded on chaotic maps and vigenere scheme which contain one round which has two iterative stages i.e. diffusion and confusion. The former step involves following stages: Forward diffusion, Matching process (using Vigenere scheme) and Backward Diffusion. G. Hanchinamani et al. in [18] proposed a scheme based on Peter De Jong chaotic map with a Rivest Cipher 4 (RC4) stream cipher. The Peter De Jong is used to control the initial keys for the RC4 stream generator. The method includes three stages: permutation (scrambling of rows and columns), pixel value rotation (circulates each pixel value by $M \times N$ random numbers) and diffusion (scan the image in two different ways). H. Liu et al. in [26] proposed an algorithm employing two-dimensional logistic map, used for the generation of keystreams, as initial parameters. An Arnold scrambling algorithm is exploited to permute the component of colour pixel. In the diffusion part authors used folding algorithm for the modification of the diffused pixel value. F. Hu et al. in [21] use the Stacked Auto Encoder (SAE). It offers parallel computing, the network generates two chaotic matrices for creation of total

Table 2 Literature survey table for image processing based image encryption algorithm

S.No.	References	Year	Type	SA	QA	DAA	OAA	RTA
1	I. Shatheesh Sam et al. [40]	2007	IP	CCA HA	MAE H	NPCR UACI	BFST KSA	×
2	X. Liao et al. [25]	2010	IP	CCA HA	×	NPCR UACI	BFST KSA	×
3	A. Akhshani et al. [3]	2012	IP	CCA HA	H	NPCR UACI	BFST	TestU01
4	I. Sam et al. [39]	2012	IP	CCA HA	H	NPCR UACI	BFST KSA	×
5	M. François et al. [16]	2012	IP	CCA HA	H	NPCR UACI	BFST KSA	NIST
6	A. A. Abd El-Latif et al. [2]	2013	IP	CCA HA	H	NPCR UACI	BFST KSA	×
7	M. SaberiKamarposhti et al. [38]	2014	IP	CCA HA	H	NPCR UACI	BFST KSA	×
8	R. Bansal et al. [8]	2014	IP	CCA HA	H MSE PSNR	NPCR UACI	BFST KSA	×
9	G. Hanchinamani et al. [18]	2015	IP	CCA HA	H MSE PSNR	NPCR UACI	BFST KSA	×
10	H. Liu et al. [26]	2017	IP	CCA HA	H	NPCR UACI	BFST KSA	×
11	F. Hu et al. [21]	2017	DL	CCA HA	×	NPCR UACI	BFST KSA	×
12	G. Ye et al. [45]	2018	IP	CCA HA	H VAR	NPCR UACI	BFST KSA	×
13	M. Kumari et al. [22]	2018	IP	CCA HA	E MSE PSNR	NPCR UACI	BFST KSA	×
14	N. Zhou et al. [47]	2018	IP	CCA HA	H MSE	×	BFST KSA NA	×
15	X. Liu et al. [27]	2019	IP	CCA HA	H MSE	×	BFST KSA NA	×
16	X. Zhang et al. [46]	2019	IP	CCA HA	H	NPCR UACI	AOA BFST KSA NA	×
17	C. He et al. [20]	2019	DL	×	×	×	×	×
18	M. Kumari et al. [23]	2020	IP	CCA HA	PSNR H	NPCR UACI	BFST CA	×
19	Y. Ding et al. [14]	2020	DL	HA	H SSIM PSNR	NPCR	BFST KSA COA CPA KPA	×
20	A. Malik et al. [31]	2021	IP	CCA HA	MSE MAE PSNR H BER	NPCR UACI	BFST NA GA AOA CST JCA CA	CST

Table 2 (continued)

S.No.	References	Year	Type	SA	QA	DAA	OAA	RTA
21	M. Alkhelaiwi et al. [5]	2021	DL	CCA	H SSIM PSNR MSE	NPCR UACI	KSA	×

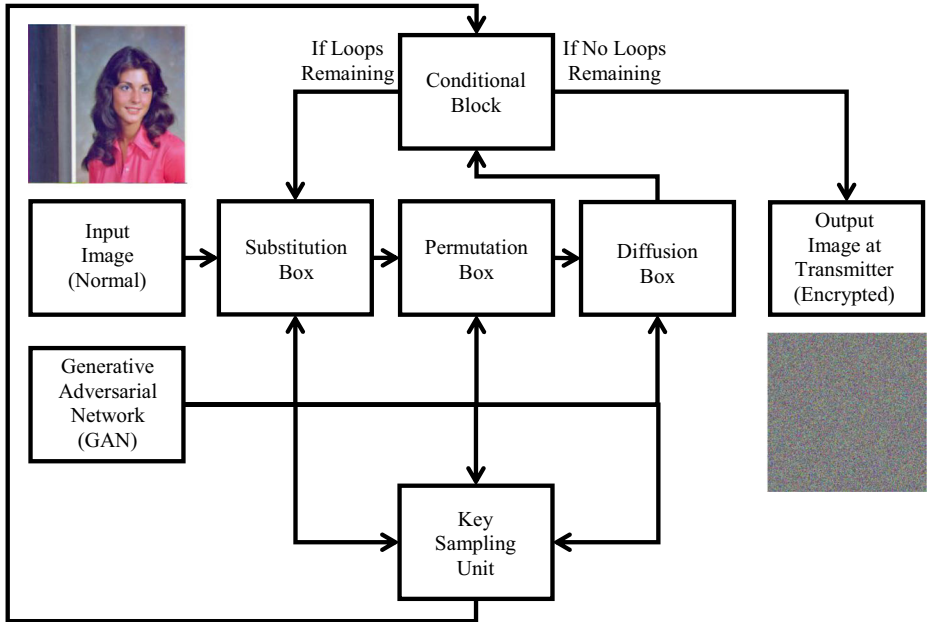
shuffling matrix and series of independence sequence, the first sequence is used to for substitution whereas the other is used for increasing the entropy by creating more confusion between the permuted and encrypted image. G. Ye et al. in [45] proposed work that included three primary operations; permutation, modulation, and diffusion. The information entropy is used for the generation of the keystream. The initial keys in the permutation and the diffusion stages communicate with each other. M. Kumari et al. in [22] proposed an image encryption scheme using intertwining chaotic maps and RC4 stream cipher for the encryption and decryption of the images. Chaotic map was used for the confusion stage and RC4 cipher for generation of key. N. Zhou et al. in [47] used the quantum cross-exchange operation and a 5-Dimension (5D) hyper chaotic system. The key feature of this scheme is employment of quantum channel swapping to swap gray level values of corresponding pixels. X. Liu et al. in [27] employed an inter-intra bit-level Permutation. Image is first converted into enhanced quantum model representation so that permutation can be applied over bit plains. For the intra-bit permutation, sorting of the chaotic sequence is done, and the inter-bit permutation is achieved with qubit operation in selected two-bit planes. Finally, diffusion is carried out with a quantum image XOR operation. X. Zhang et al. in [46] proposed an algorithms that make use of the Secure Hash Algorithm *SHA* – 256 [6, 33, 34, 36], that generates an irreversible hash sequence This sequence is then broken down into 32 equal parts containing 8 – bits each, which are used to evaluate initial values for the chaotic system. The confusion process is carried out by the employment of logistic maps that scrambles the image globally followed by block diffusion to increase the randomness in encrypted image. C. He et al. in [20] employed a Chaos-based system. The mechanism uses a deep learning network to decipher the encrypted image by first taking the image into a lower dimension space where the information of useful feature can be preserved. Once the lower dimension feature map is obtained, the map is subjected to the deconvolution generator that regenerates the similar image in same dimension as that of input image. M. Kumari et al. in [23] uses a multiple key-dependent processes that utilize these diverse keys to ensure high key sensitivity and resistance to various differential attacks. The confusion box utilizes an electronic code book, initial permutation, and bit plane scrambling followed by the diffusion process, which employed a folding technique along eight directions, exploiting different keys. Y. Ding et al. in [14] proposed DLEDNet for patient data protection in the health sector. It uses the unsupervised deep-learning network Cycle GAN for the learning part and performs image-to-image translation. The learning parameters of the transmitting side network are treated as encryption keys, whereas the learning parameters of the receiver side network are regarded as decryption keys. A. Malik et al. in [31] categorized image encryption algorithms into diverse categories and performed deep analysis, which helped to compare various image encryption algorithms on an extensive dataset and with various evaluation metrics mentioned in the upcoming sections. In M. Alkhelaiwi et al. [5], the authors make use of the Privacy Preserving Deep Learning technique. The proposed model overpowered the existing Convolution Neural Network (CNN) models in terms of diverse performance parameters.

The literature has investigated that no mechanism that optimizes all the provisions of a considerably secure method is available. Thus, digital image security for communication demands distinguished levels of protection to achieve very high brute force search time, a highly key sensitive nature, offer high entropy, a vast key space, and use a highly randomized pseudo-random number generator whose behaviour cannot be decoded easily. Also, the encrypted image must resist all kinds of attacks the intruder performs. Much work has been done on chaotic, quantum and qubit-based techniques for image encryption, but the major issue with these techniques is that they are based on a specific set of equations, and most of them do not pass various randomness tests. In recent years much work has been done in the field of deep learning, such as for solving classification problems, segmentation problems and regression problems; due to its ability to learn, the highly complex and nonlinear tasks hidden that are impossible for a human brain to process. This has motivated their application in image encryption for the last couple of years. These deep learning-based pseudo-random number generators offer highly randomized sequences after learning about various random sequences and even pass more tests than these techniques. The most important of all is that the key space drastically increases due to the addition of learning parameters of the model to exiting key space, making it impenetrable. The proposed scheme offers a highly robust mechanism based on GAN and withstands various noise and geometrical attacks. Various key features of the proposed image encryption scheme are listed below.

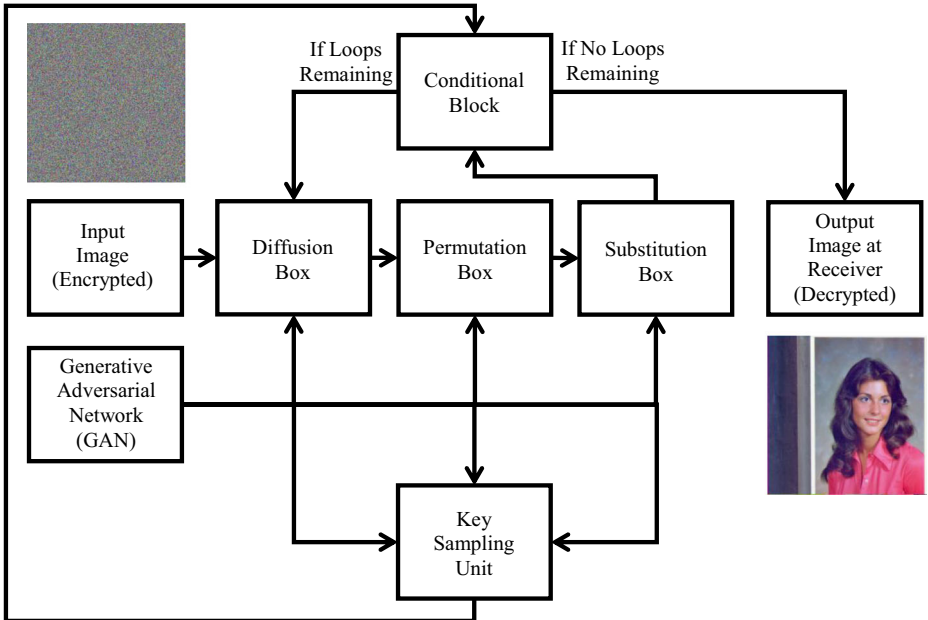
- It offers a very high brute force search time due to the use of a deep learning algorithm.
- It employs GAN an unsupervised deep learning based robust image encryption technique.
- It illustrates a high key randomness.
- The proposed scheme offers a very high key space.
- It offers byte level encryption at substitution and both byte level and bit level encryption at permutation and diffusion box.
- The proposed scheme offers a very high key sensitive mechanism for the generation of random numbers.

3 Proposed image encryption scheme

The Fig. 1 shows the complete block diagram of proposed encryption scheme. The process takes a 3-Dimension (3D) image as the input and applies a series of encryption algorithms on it. The encryption is applied at both, bit-level and byte-level to make it difficult for an intruder to decrypt the image. First an image is passed through a substitution box that is divided into two Byte-Level Substitution box i.e. Substitution Box Scheme-(A), Substitution Box Scheme-(B) followed by the Permutation box the is separated into Bit-Level Permutation Box and Byte-Level Permutation Box, these boxes adds a significant amount of confusion in the input image. Once the confusion is completed the image generated by permutation box is passed through a diffusion box that encrypts the image at both bit-level and byte-level, increasing the overall entropy of the encrypted image. The keys to substitution box, permutation box and diffusion box are generated using the trained GAN. The keys are first passed through the key sampling block that samples six sets of the generated encryption keys and drives two new keys i.e. a key for looping process and a key for selection of the folding process in diffusion box. One of the biggest advantages of using such neural network in encryption algorithm is that it provides an extra layer of security, as the intruder now requires



(a) Transmitter side encryption mechanism.



(b) Receiver side decryption mechanism

Fig. 1 Block diagram of (a) complete encryption mechanism at transmitting end, (b) complete decryption mechanism at receiving end

both the key and model in order to decipher the image. The proposed scheme also provides a mechanism to withstand the brute force attack, differential attack and other attacks available in the literature. The next subsections show the details of each block used in the block diagram.

3.1 Generative Adversarial Network (GAN)

The keys to the substitution, permutation and other blocks are given using the Generative Adversarial Network (GAN) [7, 9, 13, 17, 24, 35]. It is an unsupervised deep learning algorithm that consists of two sub-networks; a Generator model and a Discriminator model, which can be a Convolution Neural Network (CNN). As its name depicts, the generator model is responsible for generating new samples that plausibly could have come from an existing distribution of samples on which the network has learned. In contrast, a discriminator network helps a generator network to overcome its wrong learning by classifying its output into correct and false; it helps to fine-tune its and generator models performance minimizing the loss. GAN is the heart of the image encryption scheme that generates highly randomized series.

Dataset for the training GAN network is generated using the Quantum Chaotic Map explained in (Akhshani et al., 2012) [4]. For the generation of random series, these maps use some initial conditions to work on, the initial conditions in [4] are set using $x_0, y_0, z_0, x_n, y_n, z_n, r, \beta$ in eq. 1.

Quantum Chaotic Map:

$$\begin{cases} x_{n+1} = r(x_n - |x_n|^2) - ry_n, \\ y_{n+1} = -y_n e^{-2\beta} + e^{-\beta} r [(2 - x_n - x_n^*)y_n - x_n z_n^* - x_n^* z_n], \\ z_{n+1} = -z_n e^{-2\beta} + e^{-\beta} r [2(1 - x_n^*)z_n - 2x_n y_n - x_n] \end{cases} \tag{1}$$

Where,

$$\beta = \text{Dissipation parameter}$$

x_n^* and z_n^* are complex conjugate of x and z respectively

If in eq. (1) we put $y_n = 0$, and $z_n = 0$ the map results in one dimensional logistic map.

$$\{x_{n+1} = r(x_n - |x_n|^2)\} \tag{2}$$

To train GAN, two sequences are generated using the above reduced map eq. 2. To generate a sequence following steps are followed (Table 3):

In the above algorithm input $x_0 = 0.4523444338$ and $l = 10$ where x_0 is the initial condition and l is the length of numbers required to be generated using the eq. 2. First k

Table 3 Algorithm to generate training dataset for GAN

Algorithm to Generate Training Dataset

Step-1: Input x_0 and l .

Step-2: Iterate equation-(2) 1000 times and skip first k values.

Step-3: Apply $x_{n+1} = x_n \times 1000 - \text{floor}(x_n \times 1000)$.

Step-4: Repeat till l length sequence is not generated.

values of the sequence generated using eq. (2) are skipped by iterating 1000 times to reduce the transient effect produced at early iterations. In addition Step-3 is done to normalize the output. Step-2 and Step-3 are repeated until l length sequence is generated. This sequence generated is used by substitution, permutation, diffusion and key sampling process.

3.2 Substitution box

The proposed scheme's substitution box comprises two sub-boxes that perform substitution at the byte level. These boxes are called Byte-Level (A) Substitution Box and the Byte-Level (B) Substitution Box. The general block diagram is shown in Fig. 2. In the proposed scheme, the input image initially passes through the substitution box, where the pixel values are interchanged within the respective planes using eq. 3.

$$\text{New Value} = (\text{Image List}[\text{index}] + \text{KeyList}[\text{index}])\%(256) \quad (3)$$

The input image is initially passed through the first scheme, a vigenere cipher followed by the specially designed scrambling. Both boxes help in reducing the correlation among the pixels in respective planes. The output of the substitution box is highly randomized and efficiently resists visual inspection. This randomness is a result of random series generated by the trained GAN. The substitution box receives two sets of keys; the first set contains a single sequence for the Byte-Level (A) Substitution Box, and the other contains the three sequences for Byte-Level (B) Substitution Box, one for each plane.

The key size for the vigenere cipher is the same as the number of pixels in the entire image. For the image encryption process, the integer values of pixels are dealt with; thus, a slight modification is introduced in the traditional algorithm, which uses the table of integers ranging from 0 to 255 instead of A to Z. The output from this block is then passed to the next stage, where the image is scrambled plane-wise, introducing more randomness and confusion effect.

In this process, the image received by the previous block is again confused but with a different scheme. This scheme does substitution within the three image planes. For each plane, a map containing the random new values is created. Employing these maps, each pixel in the input is substituted with its new corresponding value in the map, creating a highly randomized image with better entropy and less correlation among the pixel values both plane-wise and at the aggregate level. Table 4 represents the pseudocode of the process where the image values are searched in the corresponding random map.

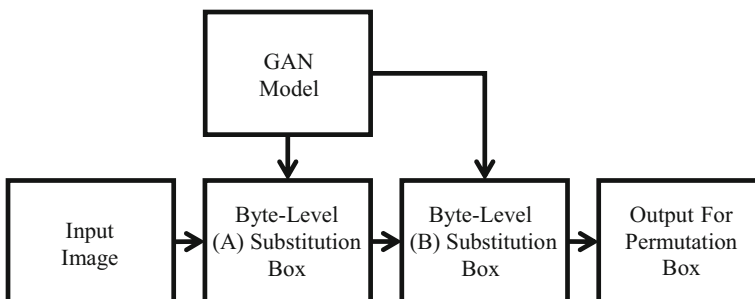


Fig. 2 General block diagram of substitution box

Table 4 Algorithm for proposed Substitution box

Algorithm of the Byte-Level (A) and (B) Substitution Box

STEP-1: INPUT RGB IMAGE AS *image* WITH SHAPE AS (*rows*, *columns*, *pages*), KEY SEQUENCE AS *key1*, THREE KEY PLANES AS *KeyForPlane0*, *KeyForPlane1* AND *KeyForPlane2* RESPECTIVELY.

STEP-2: CONVERT *image* INTO ONE DIMENSIONAL LIST AS *imageList*.

STEP-3: FOR *index* in range(0, len(*imageList*)):

CALCULATE NEW PIXEL VALUE USING equation-(3)

STEP-4: UPDATE *imageList* CURRENT POSITION PIXEL VALUE

STEP-5: END FOR

STEP-6: RESHAPE UPDATE *imageList* TO OBTAIN SUNSTITUTED IMAGE

STEP-7: FOR *i* in range(0, *pages*):

STEP-8: FOR *j* in range(0, *rows*):

STEP-9: FOR *k* in range(0, *columns*):

STEP-10: IF *i* == 0:

STEP-11: UPDATE SUNSTITUTED IMAGE PIXEL WITH *KeyForPlane0* CURRENT PIXEL VALUE

STEP-12: IF *i* == 1:

STEP-13: UPDATE SUNSTITUTED IMAGE PIXEL WITH *KeyForPlane0* CURRENT PIXEL VALUE

STEP-14: IF *i* == 2:

STEP-15: UPDATE SUNSTITUTED IMAGE PIXEL WITH *KeyForPlane0* CURRENT PIXEL VALUE

STEP-16: END IF

STEP-17: END IF

STEP-18: END IF

STEP-19: END FOR

STEP-20: END FOR

STEP-21: END FOR

STEP-22: OUTPUT RGP IMAGE IS SUBSTITUTED IMAGE AS *substitutedImage*.

Output of this algorithm is input to Permutation Box in next following section

3.3 Permutation box

The dual substitution box stage is followed by the permutation box, which performs operations at two different levels; bit-level and byte level, as shown in Fig. 3. It receives two set of encryption keys, one for bit-level permutation consists of three series containing 8 random number each, whereas the encryption keys for other block contains three planes of random numbers.

In Bit-Level Permutation Box, the image is permuted at the bit level, with the help of three random arrays, one for each plane. For the first plane, the first series having eight elements is selected. For permutation at the bit level, a pixel is selected and is converted into the binary format, in which each bit is assigned new positions based on the random series. Then the new binary is converted back to the decimal, which results in a new integer value. A similar process is carried out for the remaining two plains. This helps to generate new numbers between 0 to 255, based on repositioning the binary bits to new random positions.

The image generated by the bit-level permutation is transferred to the byte-level permutation, where the permutation occurs in the whole image to permute three maps generated by the GAN, having the same elements as that of the image. The 1st index in Plane-0, Plane-1 and Plane-2 are 2, 9 and 4 respectively, for the byte-level permutation process, the selected index is used to obtain the corresponding value in the GAN generated respective plane, this value acts

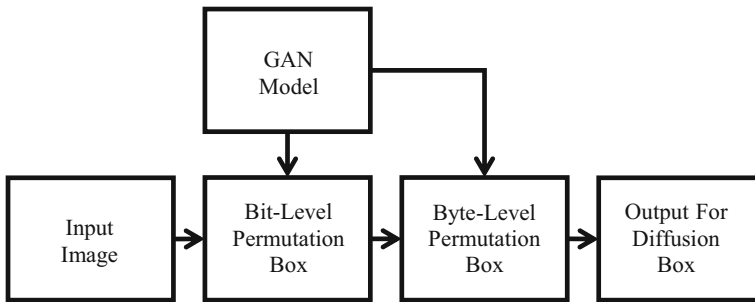


Fig. 3 General block diagram of permutation box

as the new position of the selected pixel in new image. The value is then placed at this position, to finally create the permuted image which is then send to the diffusion box. Pseudo code for the process is in the Table 5.

3.4 Diffusion box

The output of the permutation box is received by the diffusion box where the image is diffused, for eliminating any trace of originality; the box employ both levels of diffusions bit-level and byte level. Both the processes make use of *XOR* operation and two sets of random

Table 5 Algorithm for proposed permutation box

Algorithm of the Bit-Level and Byte-Level Permutation Box

STEP-1: INPUT RGB IMAGE AS substitutedImage WITH SHAPE AS (rows, columns, pages), THREE KEY SEQUENCES AS key1, key2 AND KEY3, THREE KEY PLANES AS KeyForPlane0, KeyForPlane1 AND KeyForPlane2 RESPECTIVELY. PermutedImage BE THE OUTPUT IMAGE.

STEP-2: FOR i in range(0, pages):

STEP-3: FOR j in range(0, rows):

STEP-4: FOR k in range(0, columns):

STEP-5: CALCULATE BINARY VALUE OF CURRENT PIXEL VALUE IN SubstitutedImage

STEP-6: INTERCHANGE THE POSITION OF BITS OF BINARY IN SAME BINARY CODE BASED ON CORRESPONDING PLANE WISE 8-BIT POSITION KEYS CODES.

STEP-7: CONVERT BINARY TO DECIMAL

STEP-8: UPDATE SubstitutedImage

STEP-9: END FOR

STEP-10: END FOR

STEP-11: END FOR

STEP-12: FOR i in range(0, pages):

STEP-13: FOR j in range(0, rows):

STEP-14: FOR k in range(0, columns):

STEP-15: INTERCHANGE THE POSITION OF SELECTED PIXEL IN SubstitutedImage

STEP-16: UPDATE permutedImage

STEP-17: END FOR

STEP-18: END FOR

STEP-19: END FOR

STEP-20: OUTPUT RGP IMAGE IS PERMUTED IMAGE AS permutedImage.

Output of this algorithm is input to Diffusion Box in next following section

sequences generated by GAN, in order to diffuse the image. First operation consists of three sequences carrying eight elements each, which will be used in Bit-Level Diffusion Box whereas the second set of key is used in Byte-level Diffusion and carries three planes of random elements. The results show the diffusion process generates a highly disused image that cannot be re-constructed without approved set of keys. Figure 4 shows the general block diagram of the diffusion box.

The image entered into the diffusion box, firstly undergoes the Bit-Level Diffusion, in which all the elements of each plane are converted to $8 - bit$ binary, then starting from left to right one bit is selected a time, these selected bits are then combined together to form a $3 - bit$ binary. This way, when the generated 3-Bit binary number is converted into decimal, it always results within the range of 1 to 7. This decimal is then placed in a new plane, when the same process is repeated for the remaining $7 - bits$ and pixels in three planes, total of eight new planes are generated. Before merging these planes to form the final image, *XOR* operation is performed using the sequence generated by the GAN. The 8 random numbers between 0 and 7 are used as the initial values one for each plane. The number and corresponding plane is selected and the *XOR* operation is performed element wise. The first element of the first plane is *XORed* with the random number and the resultant is stored at the selected position, now this newly generated number is *XORed* with the number present at the next position and new result is stored at the next position and so on. The same process is repeated for the remaining planes. Finally, the planes are again converted to an image by translating the binary of these planes and reconstructing the respective $8 - bit$ from $3 - bit$ binary. Once $8 - bit$ binary is formed they are placed back to respective planes forming the encrypted image. The pseudocode for the bit-level diffusion process is shown in Table 6. The output image is then passed to the Byte-Level Diffusion Box, where diffusion is done using *XORing* and Folding Process For this the box receives three highly randomized planes. The folding process basically consists of four ways; folding horizontally, vertical, along forward diagonal and backward diagonal.

This block along with the 3 randomized planes receives an extra key, which is used to identify the correct sequence of folding process. When the correct key is received the image is passed through approved combination, each sequence consists of 8 folding processes, which also perform *XORing* simultaneously. These are Horizontal Bottom Half (HBH), Horizontal Upper Half (HUH), Vertical Lower Half (VLH), Vertical Right Half (VRH), Bottom Lower Triangle (BLT), Upper Right Triangle (URT), Bottom Right Triangle (BRT) and finally Upper Left Triangle (ULT). Table 6 represents the corresponding pseudocode for the byte-level diffusion process First the selected part of the input image is *XORed* with the *KeyImage* followed by the XOR operation between the mirror pixels while performing the folding.

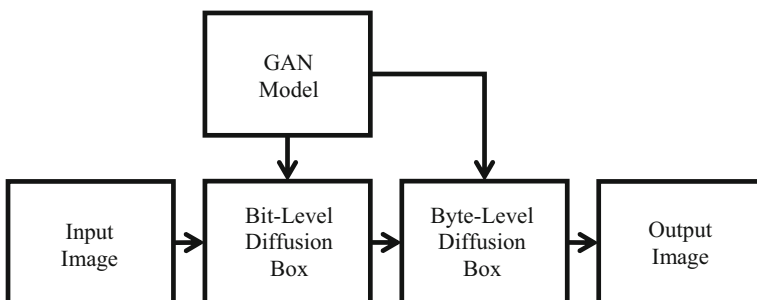


Fig. 4 Block diagram for data formulation

Table 6 Algorithm for proposed diffusion box

Algorithm of the Bit-Level and Byte-Level Diffusion Box

STEP-1: INPUT RGB IMAGE AS permutedImage WITH SHAPE AS (rows, columns, pages), KEY SEQUENCE AS key1, THREE KEY PLANES AS KeyForPlane0, KeyForPlane1 AND KeyForPlane2 RESPECTIVELY.

STEP-2: CONVERT image INTO ONE DIMENSIONAL LIST AS imageList.

STEP-3: FOR index in range(0, len(imageList)):

STEP-4 CALCULATE NEW PIXEL VALUE USING equation-(3)

STEP-5: UPDATE imageList CURRENT POSITION PIXEL VALUE

STEP-6: END FOR

STEP-7: RESHAPE UPDATE imageList TO OBTAIN SUNSTITUTED IMAGE

STEP-8: FOR i in range(0, pages):

STEP-9: FOR j in range(0, rows):

STEP-10: FOR k in range(0, columns):

STEP-11: IF i == 0:

STEP-12: UPDATE SUNSTITUTED IMAGE PIXEL WITH KeyForlane0 CURRENT PIXEL VALUE

STEP-13: IF i == 1:

STEP-14: UPDATE SUNSTITUTED IMAGE PIXEL WITH KeyForPlane0 CURRENT PIXEL VALUE

STEP-15: IF i == 2:

STEP-16: UPDATE SUNSTITUTED IMAGE PIXEL WITH KeyForPlane0 CURRENT PIXEL VALUE

STEP-17: END IF

STEP-18: END IF

STEP-19: END IF

STEP-20: END FOR

STEP-21: END FOR

STEP-22: END FOR

STEP-23: OUTPUT RGP IMAGE IS DIFFUSED IMAGE AS diffusedImage.

Output of this algorithm is final encrypted image that can be re-encrypted

4 Evaluation metrics

This section discusses evaluation metrics that is used to measure the performance and robustness of the proposed image encryption scheme. Table 7 depicts the various evaluation metrics available in the literature followed by the brief introduction to each of them.

4.1 Visual analysis (VA)

To carry out visual analysis on various encryption algorithms visual inspection is used to determine understand the encryption effect just by looking at encrypted image.

4.2 Statistical analysis (SA)

Statistical analysis collects and explores a large amount of data to discover hidden patterns and trends, for example, immediate change in contrast and structure similarity. In the image encryption schemes, the type of analysis plays an important role, as it helps to judge the relationship between the original and encrypted image, with the help of which the robustness of the scheme can be gauged. In the images, even after encryption, the neighbouring pixels exhibit some level of similarities which can be used to establish a relationship for the decoding process. To perform this analysis Correlation Coefficient (CC) and Histogram Analysis (HA) are used respectively.

Table 7 Evaluation metric used for comparison

Serial Number	Evaluation Metric	Required In	Preferred Value
1	Visual Inspection	Visual Analysis	Highly Randomizes Image
2	Correlation Coefficient	Statistical Analysis	Less
3	Histogram Analysis		
4	Mean Square Error (MSE)	Quantitative Analysis	High
5	Peak Signal to Noise Ratio (PSNR)		Low
7	Structural Similarity Index (SSIM)		Low
8	Entropy (H)		High
9	Bit Error Rate (BER)		High
10	Number of Pixels Change Rate (NPCR)	Differential Attack Analysis	Passing
11	Unified Average Changing Intensity (UACI)		Passing
12	Brute Force Search Time Attack (BFST)	Other Attack Analysis	High
13	Noise Attack		Resist
14	Geometrical Attack		Resist
15	Anti-occlusion Attack		Resist
16	Chi-Square Test		Resist
17	Crypto Analysis		Resist
18	NIST SP 800–22	Randomness Test Analysis	Passing
19	Chi-Square Test		Passing
20	Runs Test		Passing

4.3 Quantitative analysis (QA)

Quantitative analysis generally means measuring by quantity and involves exploring facts, measures, numbers, and percentages and working with numbers, statistics, formulae, and data. For this analysis Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Entropy (E), Structural Similarity Index (SSIM), and Bit Error Rate (BER) are used respectively.

4.4 Differential attack analysis (DAA)

For this analysis Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are used.

4.5 Other attack analysis (OAA)

This analysis is conducted to analyse the robustness of an image encryption algorithm against various practical attacks that may incur in communication channel. For this analysis Brute Force Search Time Attack (BFST), Noise Attack (NA), Geometrical Attack (GA), Anti-Occlusion Attack (AOA) and Crypto Analysis (CA) are used.

4.6 Randomness test analysis

Test for randomness in terms of data evaluation, is a test generally performed to analyse the distribution of a set of data to see if it can be described as random or not. For this analysis.

NIST SP 800–22, Chi-Square Test (CST) and Runs Test are used.

4.7 Simulation setup-parameters

This section discusses the detailed use of parameters used for the experimental purposes throughout.

Table 8 shows set-up parameters, used for experimentation of the proposed mechanism. This list shows the information regarding details of dataset and keys used for the scheme. These original keys are used as initial conditions for the Quantum map employed in the projected methodology and modified key values are used for testing key sensitivity of the mechanism. The specifications of the device on which implementation is performed is also detailed in the table.

5 Results and analysis

The result section showcases the various results investigated during this research and their in depth analysis. The results were computed on three set of images, each set contained 263 standard images in “.png” format. Images in Set-A, Set-B and Set-C shaped $(128 \times 128 \times 3)$, $(256 \times 256 \times 3)$, $(512 \times 512 \times 3)$ respectively.

The result section is discussed as follows **5.1 Snapshots** contains the various plots generated using the original image, encrypted image and the decrypted image for different sized inputs for once better clarity, **5.2 Visual Analysis** provides an understanding of the

Table 8 Setup-parameters

Serial Number	Parameters	Value
1	Total set used for experimentation	3
2	Total images in Set-A	263
3	Total images in Set-B	263
4	Total images in Set-C	263
5	Set-A image size used	$(128 \times 128 \times 3)$
6	Set-B image size used	$(256 \times 256 \times 3)$
7	Set-C image size used	$(512 \times 512 \times 3)$
8	Type of images used	Coloured and Grey scale images
9	Programming language version	Python-3
10	Key value (Encryption Scheme)	Original Key Values Modified Key Values
	x_0	$(128 \times 128 \times 3)$ $(128 \times 128 \times 3)$
	y_0	$x_0=0.4523444338$ $x_0'=0.4523444339$
	z_0	$y_0=0$ $y_0'=0$
	r	$z_0=0$ $z_0'=0$
	k	$r=3.9$ $r'=0.4523444339$
	<i>GenericSeed</i>	$k=10000$ $k'=10000$
	<i>KeyA</i>	<i>GenericSeed</i> = <i>GenericSeed</i> =
	<i>LoopingKey</i>	2845342442957417 4762539646935693
	<i>KeyB</i>	<i>KeyA</i> =1 <i>KeyA</i> =1
	<i>LatentSpace</i>	<i>LoopingKey</i> =1 <i>LoopingKey</i> =2
		<i>KeyB</i> =2 <i>KeyB</i> =4
		<i>LatentSpace</i> =5 <i>LatentSpace</i> =5
11	Hardware Processor	Intel(R) Xeon(R) CPU @ 2.30GHz
12	Memory RAM	13,341,992 kB (13 GB of RAM)
13	CPU Family	6
14	Model	79
15	Link for dataset used for experimentation	https://github.com/Devsingh640/Standard-Image-Dataset

robustness of the encryption scheme by visual inspection and the comparison of the results of the proposed model with the other image encryption scheme available in the literature, **5.3 Statistical Analysis** discusses the overall results based on correlation coefficient to get the insights of the image data i.e. how its pixels are now related to each other after the encryption process, **5.4 Quantitative Analysis** provides a brief comparison of the proposed scheme with other schemes available in the literature on the basis of Mean Square Error (MSE), Peak signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Entropy (H) and Bit Error Rate (BER), **5.5 Other Attack Analysis**, and finally **5.6 Randomness Test Analysis**.

5.1 Snapshots

The Table 9 presents the snapshots of the results from the proposed encryption scheme for Set-B (256, 256, 3). Starting from left to right the very first image is the original input image to the image encryption algorithm, the centre image is the encrypted image generated by the algorithms at transmitting end whereas the last image is the decrypted image at the receiver.

5.2 Visual analysis (VA)

Various outputs of different image encryption algorithm mechanisms are checked manually and verified if image data or related features are visible after encryption.

5.2.1 Visual inspection (VI)

The output of various encryption algorithm are presented in Table 10 for the visual inspection; it can be seen that no image data is visible after application of encryption algorithm; hence if the image is transmitted through the unsecured channel, the attacker will have no clue what

Table 9 Various plots generated for set-b, image shape (256,256,3.)

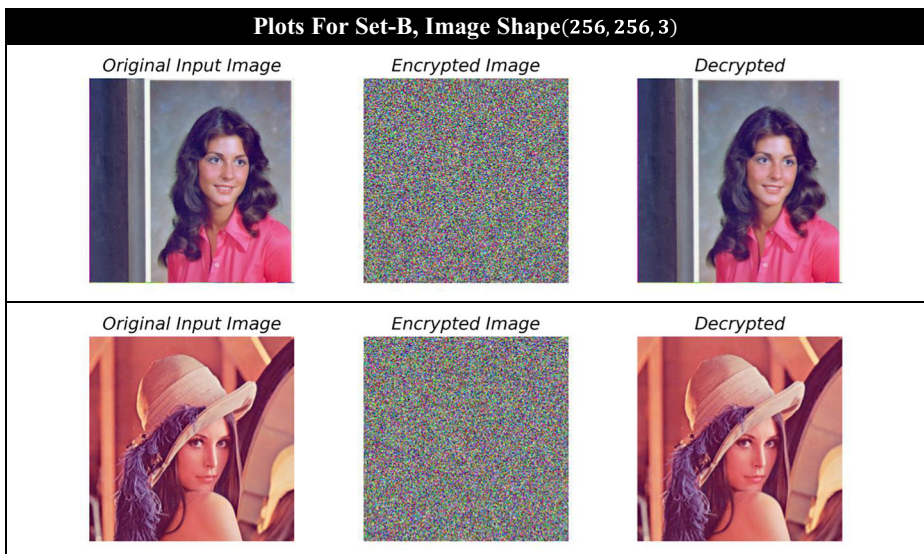

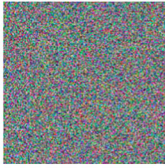








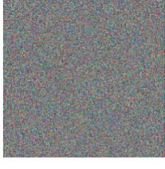




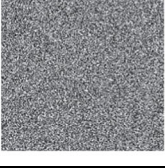


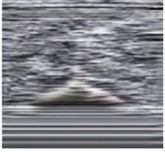




Table 10 Visual analysis table for set-b

Techniques	Encrypted Image	Techniques	Encrypted Image	Techniques	Encrypted Image
I. Shatheesh Sam <i>et al.</i> [7]		X. Liao <i>et al.</i> [8]		A. Akhshani <i>et al.</i> [9]	
I. Sam <i>et al.</i> [10]		M. François <i>et al.</i> [11]		A. A. Abd El-Latif <i>et al.</i> [12]	
M. SaberiKam arposhti <i>et al.</i> [13]		R. Bansal <i>et al.</i> [14]		G. Hanchinamani <i>et al.</i> [15]	
H. Liu <i>et al.</i> [16]		F. Hu <i>et al.</i> [17]		G. Ye <i>et al.</i> [18]	
M. Kumari <i>et al.</i> [19]		N. Zhou <i>et al.</i> [20]		X. Liu <i>et al.</i> [21]	
X. Zhang <i>et al.</i> [22]		C. He <i>et al.</i> [27]		M. Kumari <i>et al.</i> [28]	
Y. Ding <i>et al.</i> [29]		M. Alkhelaiwi <i>et al.</i> [31]		Proposed Scheme	

information the encrypted image contains. It can be seen that the proposed image encryption scheme generates a highly-secured encrypted image.

5.3 Statistical analysis (SA)

Statistical analysis collects and explores a large amount of data to discover hidden patterns and trends, for example, immediate change in contrast and structure similarity. In the image encryption schemes, the type of analysis plays an important role, as it helps to judge the relationship between the original and encrypted image, with the help of which the robustness of the scheme can be gauged. In the images, even after encryption, the neighboring pixels exhibit some level of similarities which can be used to establish a relationship for the decoding process. For a good encryption scheme, it is good to analyze the results based on these parameters like correlation coefficient and histogram analysis for evaluation and comparison.

5.3.1 Correlation coefficient (CC)

The Correlation Coefficient [2, 3, 5, 8, 12, 16, 18, 21–23, 25–27, 31, 32, 38–40, 45–47] results are computed on Set-A, Set-B and Set-C and can be studied in Tables 11, 12 and 13 respectively. The Table 14 represents the overall comparison on the basis of Correlation Coefficient for the most commonly size used in the image, i.e. Set-B. It can be clearly seen that the algorithm generated a highly uncorrelated output with respect to the original input image. Although just by looking the encrypted image in the visual analysis it seemed correlated but in fact the image is highly uncorrelated in all the directions i.e. horizontal, vertical and diagonal. It makes very difficult for the attacker to design decryption hypothesis as there is no nearly 0 correlation. The proposed image encryption scheme over powers the existing IP and DL techniques present in the literature in term of Correlation Coefficient.

$$r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)D(y)}} \quad (4)$$

5.3.2 Histogram analysis (HA)

In this analysis a frequency distribution of pixel value range shows how often each pixel value in an image occurs. A histogram [2, 3, 5, 8, 14, 16, 18, 21–23, 25–27, 31, 38–40, 45–47] is the most commonly used graph to show frequency distributions and for analysing the encryption effect. Table 15 shows histogram analysis results of Set-B (256, 256, 3).

From the above experimental results we can see that the proposed algorithm evenly distributes the frequency distribution curve resulting into approximately a straight line curve, which is highly desired by a good image encryption scheme.

5.4 Quantitative analysis (QA)

Quantitative analysis generally means to measure by quantity and involves exploring facts, measures, numbers and percentages, work with numbers, statistics, formulae and data.as the data. Here we are using quantitative analysis in order to analyse the performance of the encryption algorithm on the basis of change in image data. Methods like MSE, PSNR, SSIM, H and BER are used.

Table 11 Correlation coefficient results for the proposed algorithm on set-a

Serial Number	Parameter	OHCC	EHCC	OVCC	EVCC	ODCC	EDCC
1	Average	0.808813011	0.000691265	0.788976927	-0.000326797	0.68915426	0.000361977
2	Standard Deviation	0.182479691	0.007477163	0.235536498	0.007437522	0.274412662	0.007125557
3	Highest Value	image230.png	image25.png	image230.png	image175.png	image230.png	image177.png
4	Lowest Value	0.999495527	0.017951869	0.992549626	0.015748914	0.991938035	0.023458221
		image17.png	image232.png	image17.png	image128.png	image53.png	image129.png
		-0.048198975	-0.019278892	-0.096339295	-0.023723283	-0.23942235	-0.020325761

Table 12 Correlation coefficient results for the proposed algorithm on set-b

Serial Number	Parameter	OHCC	EHCC	OVCC	EVCC	ODCC	EDCC
1	Average	0.86928245	-0.00011485	0.85690779	0.000322871	0.776938543	-0.000221241
2	Standard Deviation	0.122668221	0.006072262	0.161898986	0.006601626	0.216269644	0.006062677
3	Highest Value	image230.png	image48.png	image111.png	image153.png	image230.png	image230.png
4	Lowest Value	0.999653321	0.019097854	0.99515015	0.02510581	0.994091697	0.000390994
		image57.png	image178.png	image6.png	image50.png	image247.png	image49.png
		0.393446147	-0.016794886	0.296204421	-0.019709475	-0.100435613	-0.017231509

Table 13 Correlation coefficient results for the proposed algorithm on set-c

Serial Number	Parameter	OHCC	EHCC	OVCC	EYCC	ODCC	EDCC
1	Average	0.914745895	0.000473998	0.90747063	0.000400257	0.855701441	0.000215113
2	Standard Deviation	0.082624168	0.005206821	0.100872754	0.005831491	0.141898729	0.005822336
3	Highest Value	image230.png	image164.png	image111.png	image154.png	image230.png	image161.png
4	Lowest Value	0.999714588 image247.png	0.017033176 image144.png	0.997567378 image247.png	0.020108717 image48.png	0.996101465 image247.png	0.020129272 image256.png
		0.459222715	-0.013320418	0.448470518	-0.015919477	-0.022065706	-0.01695511

Table 14 Correlation coefficient results, comparison results with techniques available in the literature on set-b

Serial Number	Techniques	EHCC	EVCC	EDCC
1	I. Shatheesh Sam et al. [40]	-0.02854	-0.01203	0.017213
2	X. Liao et al. [25]	-0.0014166	-0.0127833	0.0082166
3	A. Akhshani et al. [3]	-0.04141	-0.01499	-0.00097
4	I. Sam et al. [39]	0.035828	0.007072	0.022069
5	M. François et al. [16]	-0.0136	-0.00149	-0.02528
6	A. A. Abd El-Latif et al. [2]	-0.02267	-0.02488	0.011189
7	M. SaberiKamarposhti et al. [38]	-0.0059	0.0056	0.0087
8	R. Bansal et al. [8]	-0.00083	-0.00088	0.011412
9	G. Hanchinamani et al. [18]	0.001197	0.013619	-0.01739
10	H. Liu et al. [26]	-0.01565	-0.01766	0.006531
11	F. Hu et al. [21]	-0.0049625	0.0034375	0.009775
12	G. Ye et al. [45]	-0.00721	-0.00502	-0.00974
13	M. Kumari et al. [22]	-0.0050	-0.0089	-0.0124
14	N. Zhou et al. [47]	-0.01445	0.018132	0.011508
15	X. Liu et al. [27]	0.023656	-0.02194	0.001439
16	X. Zhang et al. [46]	-0.0022333	0.0007666	-0.0004333
17	C. He et al. [20]	-	-	-
18	M. Kumari et al. [23]	0.00506	0.01858	-0.019313
19	Y. Ding et al. [14]	-	-	-
20	M. Alkhelaiwi et al. [5]	-0.0039931	0.0024391	-0.0041093
21	Proposed Scheme	-0.00011485	0.000322871	-0.000221241

Table 15 Various plots generated for set-b, image shape (256,256,3)

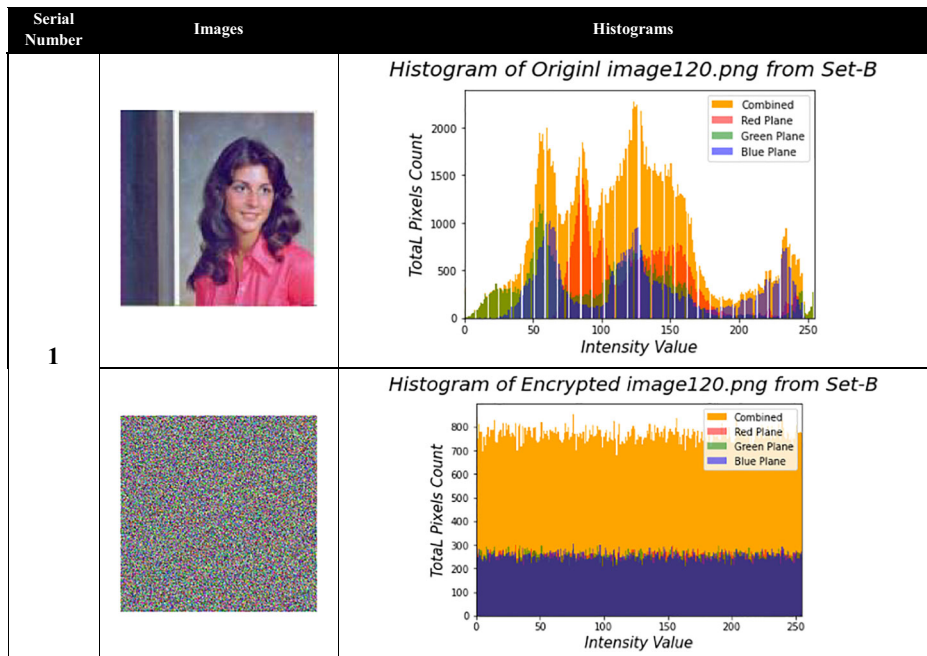


Table 16 Mean square error results for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	MSE (128,128,3)	MSE (256,256,3)	MSE (512,512,3)
1	Average	9029.355793	9262.928652	9414.945997
2	Standard Deviation	2163.861495	2210.898431	2310.811892
3	Highest Value	image136.png 20,061.45542	image136.png 20,866.93925	image247.png 21,739.14812
4	Lowest Value	image52.png 5816.188253	image52.png 5881.719243	image52.png 5911.809263

5.4.1 Mean square error (MSE)

Table 16 represents the MSE [5, 8, 18, 22, 27, 31, 47] results computed on Set-A, Set-B, and Set-C. Table 17 represents the overall comparison on the basis of MSE for available techniques in the literature. The average MSE value for the proposed scheme is 9262.928652 that is a high than most of the techniques available literature. For given two images of dimension (i, j, k) , $MSE(O, E)$ can be calculated according formula as given below:

$$MSE(O, E) = \left(\frac{\sum_k \sum_i \sum_j (O(i, j, k) - E(i, j, k))^2}{N} \right) \times 100 \tag{5}$$

Table 17 Mean Square Error, results comparison results with techniques available in the literature on set-b

Serial Number	Technique Name	Experimental Values	Standard Deviation
1	I. Shatheesh Sam et al. [40]	125.5988	2.293259
2	X. Liao et al. [25]	–	–
3	A. Akhshani et al. [3]	120.5384	2.522516
4	I. Sam et al. [39]	124.3401	2.284503
5	M. François et al. [16]	127.3067	2.346352
6	A. A. Abd El-Latif et al. [2]	126.323	2.307419
7	M. SaberiKamarposhti et al. [38]	–	–
8	R. Bansal et al. [8]	130.6876	3.519312
9	G. Hanchinamani et al. [18]	121.8786	2.648096
10	H. Liu et al. [26]	123.3918	1.90986
11	F. Hu et al. [21]	–	–
12	G. Ye et al. [45]	125.4028	1.107805
13	M. Kumari et al. [22]	128.6887	20.43007
14	N. Zhou et al. [47]	124.8899	9.878425
15	X. Liu et al. [27]	125.8735	10.22355
16	X. Zhang et al. [46]	–	–
17	C. He et al. [20]	–	–
18	M. Kumari et al. [23]	109.0863	26.35046
19	Y. Ding et al. [14]	18.98805	–
20	M. Alkhelaiwi et al. [5]	21,235.88	–
21	Proposed Scheme	9262.928652	2210.898431

Table 18 Peak signal to noise ratio results for the proposed algorithm on set-a, set-b and set-c

Serial Number	Parameter	PSNR (128,128,3)	PSNR (256,256,3)	PSNR (512,512,3)
1	Average	8.680528561	8.568832147	8.503338709
2	Standard Deviation	0.932611605	0.930130241	0.951916387
3	Highest Value	image52.png 10.48441906	image52.png 10.43576071	image52.png 10.41359947
4	Lowest Value	image136.png 5.107179237	image136.png 4.936216091	image247.png 4.758378393

5.4.2 Peak signal to noise ratio (PSNR)

Table 18 represents the PSNR [5, 8, 12, 14, 18, 22, 31, 32, 42, 43, 45] results computed on Set-A, Set-B, and Set-C.

Table 19 represents the overall comparison on the basis of PSNR for available techniques in the literature. The average PSNR value for the proposed scheme is 8.568832147 which is lower than most of the techniques available literature.

$$PSNR(O, E) = 10 \times \log_{10} \left(\frac{(I_{max})^2}{MSE(O, E)} \right) \tag{6}$$

Table 19 Peak signal to noise ratio results, comparison results with techniques available in the literature on set-b

Serial Number	Technique Name	Experimental Values	Standard Deviation
1	I. Shatheesh Sam et al. [40]	27.14095	0.080029
2	X. Liao et al. [25]	–	–
3	A. Akhshani et al. [3]	27.31955	0.0918496
4	I. Sam et al. [39]	27.18469	0.0805351
5	M. François et al. [16]	27.08229	0.0807903347
6	A. A. Abd El-Latif et al. [2]	27.11598	0.0800618
7	M. SaberiKamarposhti et al. [38]	–	–
8	R. Bansal et al. [8]	26.96846	0.1185556
9	G. Hanchinamani et al. [18]	27.27153	0.0954008
10	H. Liu et al. [26]	27.21794	0.0677458
11	F. Hu et al. [21]	–	–
12	G. Ye et al. [45]	27.14773	0.0385359
13	M. Kumari et al. [22]	27.0354	0.750779
14	N. Zhou et al. [47]	27.16553	0.357861402
15	X. Liu et al. [27]	27.13146	0.367888603
16	X. Zhang et al. [46]	–	–
17	C. He et al. [20]	–	–
18	M. Kumari et al. [23]	27.7531	1.200765
19	Y. Ding et al. [14]	35.346	–
20	M. Alkhelaiwi et al. [5]	4.8601	–
21	Proposed Scheme	8.568832147	0.930130241

5.4.3 Entropy (H)

Table 20 represents the entropy [2, 3, 8, 16, 18, 22, 23, 26, 27, 31, 32, 38–40, 45–47] results computed on Set-A, Set-B, and Set-C. Table 21 represents the overall comparison on the basis of entropy for available techniques in the literature. As inferred from table, the average value for the proposed scheme is 7.999057982, that is a very high average value in comparison to the results of the available mechanisms in literature with a reduced amount of standard deviation.

$$H(x) = - \sum_{i=1}^n P(x_i) \log_2 P(x_i) \quad (7)$$

5.4.4 Structural similarity index (SSIM)

Structural Similarity Index (SSIM) [5, 10, 14, 15, 41, 43] is one of the a robust technique for the measurement of image fidelity, according to the basic principal of SSIM for image fidelity measurement the retention of signal structure is of importance [42]. SSIM measures the index value based on luminance value, contrast value and finally structure calculated at patch level. Table 22 represents the SSIM results computed on Set-A, Set-B and Set-C. This depicts the original input image and encrypted image has no structural similarities in common. This is highly recommended by a good image encryption technique.

$$S(x, y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \right) \quad (8)$$

5.4.5 Bit error rate (BER)

The bit error rate [31] is defined as the rate at which errors occur in a transmission system. In image encryption point of view it is a very important parameter in characterising the performance of data channels. Table 23 represents the BER results computed on Set-A, Set-B and Set-C. This depicts the BER among the original input image and encrypted image for the proposed image encryption technique.

5.5 Differential attack analysis (DAA)

The results for the differential attack analysis are as shown in the Tables 24, 25, 26, 27, 28, 29, 30 and 31 it can be seen that the proposed image encryption scheme is able to pass both NPCR and UACI test.

5.5.1 Number of pixels change rate (NPCR)

Number of Pixels Change Rate [2, 3, 5, 8, 12, 14, 16, 18, 21–23, 25, 26, 31, 32, 38–40, 44–46] usually abbreviates as NPCR is one of the criteria used to analyse the differential attack. NPCR means the change rate of the number of pixels in ciphered image when one pixel of the original-image is modified. Table 27 depicts the results for NPCR test, and it can be seen that the proposed image encryption schemes robust in nature against attacks,

Table 20 Entropy results for the proposed algorithm on set-a, set-b and set-c

Serial Number	Parameter	EOI (128,128,3)	EI (128,128,3)	EOI (256,256,3)	EI (256,256,3)	EOI (512,512,3)	EI (512,512,3)
1	Average	6.903079262	7.996257522	6.919105557	7.999057982	6.85277355	7.999767152
2	Standard Deviation	0.986549169	0.000308979	1.056016768	7.59159E-05	1.125991445	2.16605E-05
3	Highest Value	image63.png	image78.png	image247.png	image135.png	image247.png	image212.png
4	Lowest Value	7.901498761	7.996964125	7.978884476	7.999278727	7.933743684	7.999822306
		image247.png	image16.png	image26.png	image169.png	image223.png	image236.png
		1.374032227	7.995413649	0.868749406	7.998808378	0.500032851	7.999707918

Table 21 Entropy results, comparison results with techniques available in the literature on set-b

Serial Number	Technique Name	Experimental Values	Standard Deviation
1	I. Shatheesh Sam et al. [40]	7.985093	0.000392
2	X. Liao et al. [25]	–	–
3	A. Akhshani et al. [3]	7.950513	0.002898
4	I. Sam et al. [39]	7.985027	0.000724
5	M. François et al. [16]	7.985436	0.0003544
6	A. A. Abd El-Latif et al. [2]	7.983825	0.002196
7	M. SaberiKamarposhti et al. [38]	7.9919	–
8	R. Bansal et al. [8]	7.999523	0.006474
9	G. Hanchinamani et al. [18]	7.962831	0.000903
10	H. Liu et al. [26]	7.985556	0.152205
11	F. Hu et al. [21]	–	–
12	G. Ye et al. [45]	7.953657	0.02561
13	M. Kumari et al. [22]	7.984129	0.000733
14	N. Zhou et al. [47]	7.974637	0.151696177
15	X. Liu et al. [27]	7.956454	0.022862846
16	X. Zhang et al. [46]	7.99765	–
17	C. He et al. [20]	–	–
18	M. Kumari et al. [23]	7.999009	0.00011
19	Y. Ding et al. [14]	7.955	–
20	M. Alkhelaiwi et al. [5]	7.95	–
21	Proposed Scheme	7.999057982	7.59159E-05

as the average value is above the theoretical NPCR critical values for all levels which is essential to pass the test.

$$NPCR(OC, MC) = \frac{\sum_k \sum_i \sum_j |Sign(OE(i, j, k) - ME(i, j, k))|}{N} \times 100\% \tag{9}$$

5.5.2 Unified average changing intensity (UACI)

Unified Average Changing Intensity [2, 3, 5, 8, 12, 16, 18, 21–23, 25, 26, 31, 32, 38–40, 44–46] usually abbreviated as UACI is the other criteria used to analyse the deferential attack, UACI calculates the average intensity of ciphered image when one pixel of the original-image is modified. Table 31 depicts the results for UACI test, and it can be seen that the proposed

Table 22 Structural Similarity Index results for the proposed algorithm on set-a, set-b and set-c

Serial Number	Parameter	SSIM (128,128,3)	SSIM (256,256,3)	SSIM (512,512,3)
1	Average	0.008571264	0.008636546	0.008680341
2	Standard Deviation	0.002457181	0.001673684	0.0014192
3	Highest Value	image23.png 0.014876596	image46.png 0.012345506	image214.png 0.010751197
4	Lowest Value	image252.png 0.00043632	image24.png 0.002440628	image146.png 0.002825159

Table 23 Bit error rate results for the proposed algorithm on set-a, set-b and set-c

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.3320	0.3320	0.3320
2	Standard Deviation	9.5653e-05	4.6934e-05	2.1813e-05
3	Highest Value	Image115.png 0.3323	image178.png 0.3321	image159.png 0.3321
4	Lowest Value	Image132.png 0.3316	image66.png 0.3319	image253.png 0.3319

Table 24 NPCR test results for the proposed algorithm on set-a

Serial Number	Parameter	Experimental Values	Theoretical NPCR Critical Value		
			$N_{0.05}^* = 99.5292$ Level 0.05	$N_{0.01}^* = 99.4960$ Level 0.01	$N_{0.001}^* = 99.4588$ Level 0.001
1	Average	99.61087574	Pass	Pass	Pass
2	Standard Deviation	0.025747845	–	–	–
3	Highest Value	image35.png 99.69482422	Pass	Pass	Pass
4	Lowest Value	image149.png 99.56054688	Pass	Pass	Pass

image encryption schemes robust in nature against attacks as the average value is between the theoretical UACI critical values for all levels which is essential to pass the test.

$$UACI(OC, MC) = \frac{\sum_k \sum_i \sum_j |OC(i, j, k) - MC(i, j, k)|}{N \times 255} \times 100\% \tag{10}$$

Table 25 NPCR test results for the proposed algorithm on set-b

Serial Number	Parameter	Experimental Values	Theoretical NPCR Critical Value		
			$N_{0.05}^* = 99.5693$ Level 0.05	$N_{0.05}^* = 99.5693$ Level 0.01	$N_{0.05}^* = 99.5693$ Level 0.001
1	Average	99.60947943	Pass	Pass	Pass
2	Standard Deviation	0.01373395	–	–	–
3	Highest Value	image10.png 99.60123698	Pass	Pass	Pass
4	Lowest Value	image21.png 99.56970215	Pass	Pass	Pass

Table 26 NPCR test results for the proposed algorithm on Set-C

Serial Number	Parameter	Experimental Values	Theoretical NPCR Critical Value		
			$N_{0.05}^* = 99.5893$ Level 0.05	$N_{0.01}^* = 99.5810$ Level 0.01	$N_{0.001}^* = 99.5717$ Level 0.001
1	Average	99.60880159	Pass	Pass	Pass
2	Standard Deviation	0.006963712	–	–	–
3	Highest Value	image6.png 99.62654114	Pass	Pass	Pass
4	Lowest Value	image38.png 99.59373474	Pass	Pass	Pass

5.6 Other attack analysis (OAA)

These attacks are generally carried out to ensure the good performance of the image encryption algorithm in practical environment as, in practical application the encrypted is exposed to various kinds of distortions caused due to channel noise or attacker etc. it is very important to test an algorithm against these attacks, it consists of key space analysis, noise attack, geometry

Table 27 NPCR results, comparison results with techniques available in the literature on Set-B

Serial Number	Technique Name	Experimental Values	Theoretical NPCR Critical Value		
			$N_{0.05}^* = 99.5693$ 0.05 Level	$N_{0.01}^* = 99.5527$ 0.01 Level	$N_{0.001}^* = 99.5341$ 0.001 Level
1	I. Shatheesh Sam et al. [40]	99.61344%	Pass	Pass	Pass
2	X. Liao et al. [25]	99.65%	Pass	Pass	Pass
3	A. Akhshani et al. [3]	99.612426%	Pass	Pass	Pass
4	I. Sam et al. [39]	99.568176%	Fail	Pass	Pass
5	M. François et al. [16]	99.628194%	Pass	Pass	Pass
6	A. A. Abd El-Latif et al. [2]	99.5513%	Fail	Fail	Pass
7	M. SaberiKamarposhti et al. [38]	99.68886	Pass	Pass	Pass
8	R. Bansal et al. [8]	99.61299%	Pass	Pass	Pass
9	G. Hanchinamani et al. [18]	99.624633%	Pass	Pass	Pass
10	H. Liu et al. [26]	99.59971%	Pass	Pass	Pass
11	F. Hu et al. [21]	99.5538	Fail	Pass	Pass
12	G. Ye et al. [45]	99.594416%	Pass	Pass	Pass
13	M. Kumari et al. [22]	99.6012369%	Pass	Pass	Pass
14	N. Zhou et al. [47]	51.2329%	Fail	Fail	Fail
15	X. Liu et al. [27]	50.2025%	Fail	Fail	Fail
16	X. Zhang et al. [46]	99.6269	Pass	Pass	Pass
17	C. He et al. [20]	–	–	–	–
18	M. Kumari et al. [23]	99.611218%	Pass	Pass	Pass
19	Y. Ding et al. [14]	94.21%	Fail	Fail	Fail
20	M. Alkhelaiwi et al. [5]	99.5725%	Pass	Pass	Pass
21	Proposed Scheme	99.60947943	Pass	Pass	Pass

Table 28 UACI test results for the proposed algorithm on set-a

Serial Number	Parameter	Experimental Values	Theoretical UACI Critical Value		
			$N_{0.05}^{*-}$ = 33.1012 Level 0.05	$N_{0.01}^{*-}$ = 32.9874 Level 0.01	$N_{0.001}^{*-}$ = 32.8552 Level 0.001
1	Average	33.47411609	Pass	Pass	Pass
2	Standard Deviation	0.184381011	–	–	–
3	Highest Value	image249.png 33.8229789	Pass	Pass	Pass
4	Lowest Value	image110.png 33.12456916	Pass	Pass	Pass

Table 29 UACI test results for the proposed algorithm on set-b

Serial Number	Parameter	Experimental Values	Theoretical UACI Critical Value		
			$N_{0.05}^{*-}$ = 33.2824 Level 0.05	$N_{0.01}^{*-}$ = 33.2255 Level 0.01	$N_{0.001}^{*-}$ = 33.1594 Level 0.001
1	Average	33.45799836	Pass	Pass	Pass
2	Standard Deviation	0.089420135	–	–	–
3	Highest Value	image218.png 33.64387063	Pass	Pass	Pass
4	Lowest Value	image156.png 33.28537286	Pass	Pass	Pass

Table 30 UACI test results for the proposed algorithm on set-c

Serial Number	Parameter	Experimental Values	Theoretical UACI Critical Value		
			$N_{0.05}^{*-}$ = 33.3730 Level 0.05	$N_{0.01}^{*-}$ = 33.3445 Level 0.01	$N_{0.001}^{*-}$ = 33.3115 Level 0.001
1	Average	33.46615389	Pass	Pass	Pass
2	Standard Deviation	0.047961912	–	–	–
3	Highest Value	image136.png 33.54916741	Pass	Pass	Pass
4	Lowest Value	image255.png 33.37824803	Pass	Pass	Pass

Table 31 UACI results, comparison results with techniques available in the literature on set-b

Serial Number	Technique Name	Experimental Values	Theoretical UACI Critical Value		
			$N_{0.05}^{*-}$ = 33.284 0.05 Level	$N_{0.01}^{*-}$ = 33.2255 0.01 Level	$N_{0.001}^{*-}$ = 33.1594 $N_{0.05}^{*+}$ = 33.7677 0.001 Level
1	I. Shatheesh Sam et al. [40]	33.4598%	Pass	Pass	Pass
2	X. Liao et al. [25]	0.3348	Pass	Pass	Pass
3	A. Akhshani et al. [3]	33.5012%	Pass	Pass	Pass
4	I. Sam et al. [39]	33.4713%	Pass	Pass	Pass
5	M. François et al. [16]	33.4688%	Pass	Pass	Pass
6	A. A. Abd El-Latif et al. [2]	33.4612%	Pass	Pass	Pass
7	M. SaberiKamarposhti et al. [38]	33.24052%	Fail	Pass	Pass
8	R. Bansal et al. [8]	33.37299%	Pass	Pass	Pass
9	G. Hanchinamani et al. [18]	33.5468%	Pass	Pass	Pass
10	H. Liu et al. [26]	33.5638%	Pass	Pass	Pass
11	F. Hu et al. [21]	33.59315%	Pass	Pass	Pass
12	G. Ye et al. [45]	33.4783%	Pass	Pass	Pass
13	M. Kumari et al. [22]	33.702863%	Pass	Pass	Pass
14	N. Zhou et al. [47]	33.4674%	Pass	Pass	Pass
15	X. Liu et al. [27]	25.0907%	Fail	Fail	Fail
16	X. Zhang et al. [46]	33.5461%	Pass	Pass	Pass
17	C. He et al. [20]	–	–	–	–
18	M. Kumari et al. [23]	33.4942%	Pass	Pass	Pass
19	Y. Ding et al. [14]	–	–	–	–
20	M. Alkhelaiwi et al. [5]	33.66%	Fail	Pass	Pass
21	Proposed Scheme	33.45799836	Pass	Pass	Pass

attack anti occlusion attack etc. Tables 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 and 57 and depicts these results.

5.6.1 Brute force search time attack (BFST)

In this attack an attacker tries to use all permutation and combination of keys unless he gets the key that decrypts the encrypted image. This process takes a huge amount of time, due to which it this process is also known as exhaustive key search. Since the time is important to break any cipher image is relative to the span of the secret key in this we test the key size. [2, 3, 5, 8, 14, 16, 18, 21–23, 25–27, 31, 38–40, 45–47]. Table 32 shows the key space for the various techniques available in the literature techniques for the comparison. It can be seen that techniques based on quantum, chaos and qubit have a pretty good key space but the machine learning based techniques offers a very high amount of brute force search time due to the addition of network learnable parameters making it impossible to crack.

5.6.2 Noise attack (NA)

Encrypted data when sent through an open channel is highly vulnerable and even experience get noisy, that result in problem while decryption. The most common noise introduced to an

Table 32 Brute force search time attack results comparison results with techniques available in the literature

Serial Number	Technique Name	Key Space
1	I. Shatheesh Sam et al. [40]	$2^{192} - 2^{216}$
2	X. Liao et al. [25]	2^{128}
3	A. Akhshani et al. [3]	2^{256}
4	I. Sam et al. [39]	2^{192}
5	M. François et al. [16]	2^{462}
6	A. A. Abd El-Latif et al. [2]	2^{224}
7	M. SaberiKamarposhti et al. [38]	2^{120}
8	R. Bansal et al. [8]	2^{448}
9	G. Hanchinamani et al. [18]	2^{384}
10	H. Liu et al. [26]	2^{128}
11	F. Hu et al. [21]	10^{21}
12	G. Ye et al. [45]	2^{42}
13	M. Kumari et al. [22]	2^{384}
14	N. Zhou et al. [47]	2^{72}
15	X. Liu et al. [27]	$>2^{100}$
16	X. Zhang et al. [46]	$3.4028 * 1098$
17	C. He et al. [20]	<i>key-independent (Network Parameters)</i>
18	M. Kumari et al. [23]	2^{432}
19	Y. Ding et al. [14]	$(10^{10})^{2757936}$
20	M. Alkhelaiwi et al. [5]	$(10^{10})^{10,77,540}$
21	Proposed Scheme	$>(10^{10})^{10,77,540}$

image when passed through the noisy channel are salt & pepper, Poisson, Gaussian and speckle noise. An image encryption algorithm must with stand these types of noises so that information could be transferred properly. [27, 31, 46, 47]

In this attach the encrypted image is exposed to salt and pepper noise in order to test the robustness of the algorithm against noise, for this experiment a total of 10 % , 25%and 50% pixels of single plane are treated as the noise and noise is added to these pixels in all three planes of the encrypted image and finally decrypted. The results in Table 33 show how good the algorithm performed on such huge amount of channel added noise.

5.6.3 Geometrical attack (GA)

Geometrical Attack [31] is basically geometric distortion caused in an image intentionally or unintentionally, these may be due to rotation or flipping the original image. To test the algorithm against the known as de-synchronization attack various flip and rotation functions are applied on the encrypted image before sending it could be transmitted through the channel, a good algorithm is capable of recovering some portion of original input image at the receiver side this is because geometric distortion like these in an image make it difficult and sometimes impossible to identify the original data. Tables 44, 45, 46, 47, 48, 49, 50, 51, 52 and 53 shows the different flips and rotation attacks used for the experimentation.

Flip attack In flip attack the encrypted image is flipped before transmitting, the different type of flip attacks can be seen in Table 44. The proposed algorithm withstands the flip attack and overpowers the available techniques in literature as seen in Tables 48 and 49 in term of PSNR and BER.

Table 33 Various plots generated for encrypted image corrupted by Salt and Pepper noise with amount 10 % , 25%and 50% respectively





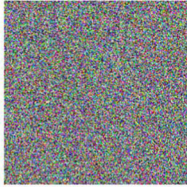


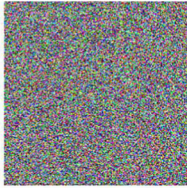


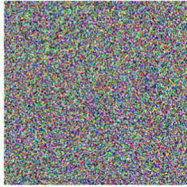
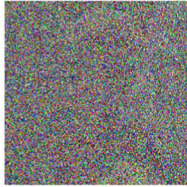
Plots For Set-B, Image Shape (256, 256, 3)		
<i>Original Input Image</i> 	<i>Encrypted Image</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Salt & Pepper 10% Flip Attack Applied</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Salt & Pepper 25% Flip Attack Applied</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Salt & Pepper 50% Flip Attack Applied</i> 	<i>Decrypted</i> 

Table 34 Different types of flip operations on an image used in the analysis of geometrical attack





OII	VFI	HFI	VHFI
			

Table 35 BER results after vertical flipped attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.249085	0.332029	0.249029
2	Standard Deviation	0.000084	0.000050	0.000022
3	Highest Value	image213.png 0.249376	image174.png 0.332182	image168.png 0.249083
4	Lowest Value	image14.png 0.248779	image69.png 0.331904	image28.png 0.248965

Table 36 BER results after horizontal flipped attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.169891	0.332030	0.166987
2	Standard Deviation	0.000059	0.000053	0.000016
3	Highest Value	image261.png 0.170071	image178.png 0.032618	image246.png 0.167037
4	Lowest Value	image178.png 0.169698	image261.png 0.331868	image21.png 0.166925

Rotation attack In rotation attack the encrypted image is rotated before transmitting, the different rotation attacks can be seen in Table 41. Tables 42, 43 and 44 demonstrate BER and PSNR values for all the SETs of Dataset. The proposed algorithm is withstands the rotation attack and overpowers the available techniques in literature as seen in Tables 45 and 46 in term of PSNR and BER. Table 47 shows various plots generated for encrypted image corrupted by Rotation Attack on Set-C.

5.6.4 Anti-occlusion attack (AOA)

To analyse the anti-occlusion [31, 46] capability of the proposed algorithm against the loss of data, the same encrypted image is occluded with $1/64, 1/16, 1/4, 1/2$ & $3/4$ respectively. Then, the blocked images are decrypted with the algorithm with which it is encrypted. This attack is usually done to test the algorithms performance when there is some loss in the data, to carry out this test $1/64, 1/16, 1/4, 1/2, 3/4$ occlusion schemes are used as shown in Table 48. The

Table 37 BER results after vertical then horizontal flipped attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.250374	0.332038	0.249352
2	Standard Deviation	0.000077	0.000046	0.000021
3	Highest Value	image155.png 0.250583	image255.png 0.332182	image57.png 0.249410
4	Lowest Value	image24.png 0.250142	image151.png 0.331911	image135.png 0.249290

Table 38 PSNR results, comparison results for horizontal flip attack with techniques available in the literature on set-b

Serial Number	Technique Name	VFI	HFI	VHFI
1	I. Shatheesh Sam et al. [40]	–	27.15986	–
2	X. Liao et al. [25]	–	–	–
3	A. Akhshani et al. [3]	–	25.67547	–
4	I. Sam et al. [39]	–	27.69424	–
5	M. François et al. [16]	–	27.37505	–
6	A. A. Abd El-Latif et al. [2]	–	27.43363	–
7	M. SaberiKamarposhti et al. [38]	–	–	–
8	R. Bansal et al. [8]	–	26.96252	–
9	G. Hanchinamani et al. [18]	–	27.15986	–
10	H. Liu et al. [26]	–	25.67547	–
11	F. Hu et al. [21]	–	–	–
12	G. Ye et al. [45]	–	27.37505	–
13	M. Kumari et al. [22]	–	26.91006	–
14	N. Zhou et al. [47]	–	27.24573	–
15	X. Liu et al. [27]	–	25.67547	–
16	X. Zhang et al. [46]	–	–	–
17	C. He et al. [20]	–	–	–
18	M. Kumari et al. [23]	–	27.38073	–
19	Y. Ding et al. [14]	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–
21	Proposed Scheme	8.569768522	8.569485942	8.568771017

Table 39 BER results, comparison results for flip attack with techniques available in the literature on Set-B

Serial Number	Technique Name	VFI	HFI	VHFI
1	I. Shatheesh Sam et al. [40]	–	0.996175	–
2	X. Liao et al. [25]	–	–	–
3	A. Akhshani et al. [3]	–	0.992839	–
4	I. Sam et al. [39]	–	0.995605	–
5	M. François et al. [16]	–	0.994141	–
6	A. A. Abd El-Latif et al. [2]	–	0.993815	–
7	M. SaberiKamarposhti et al. [38]	–	–	–
8	R. Bansal et al. [8]	–	0.996112	–
9	G. Hanchinamani et al. [18]	–	0.996175	–
10	H. Liu et al. [26]	–	0.992839	–
11	F. Hu et al. [21]	–	–	–
12	G. Ye et al. [45]	–	0.994141	–
13	M. Kumari et al. [22]	–	0.996826	–
14	N. Zhou et al. [47]	–	0.996826	–
15	X. Liu et al. [27]	–	0.992839	–
16	X. Zhang et al. [46]	–	–	–
17	C. He et al. [20]	–	–	–
18	M. Kumari et al. [23]	–	0.993589	–
19	Y. Ding et al. [14]	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–
21	Proposed Scheme	0.332029	0.332030	0.332038

Table 40 Various plots generated for encrypted image corrupted by flip attack on set-c








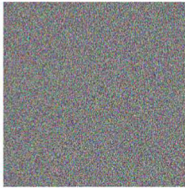


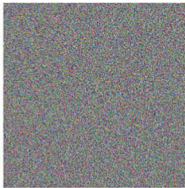

Plots For Set-C, Image Shape (512, 512, 3)		
<p><i>Original Input Image</i></p> 	<p><i>Encrypted Image</i></p> 	<p><i>Decrypted</i></p> 
<p><i>Original Input Image</i></p> 	<p><i>Vertical Flip Attack Applied</i></p> 	<p><i>Decrypted</i></p> 
<p><i>Original Input Image</i></p> 	<p><i>Horizontal Flip Attack Applied</i></p> 	<p><i>Decrypted</i></p> 
<p><i>Original Input Image</i></p> 	<p><i>Vertical-Horizontal Flip Attack Applied</i></p> 	<p><i>Decrypted</i></p> 

Table 41 Different types of rotation operations on an image used in the analysis of geometrical attack





OII	CWR-90°	CWR-180°	ACWR-90°
			

Table 42 BER results after clock wise 90° Rotation attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.332028	0.332032	0.332031
2	Standard Deviation	0.000095	0.000048	0.000024
3	Highest Value	image92.png 0.332357	image69.png 0.332148	image46.png 0.332105
4	Lowest Value	image44.png 0.331746	image101.png 0.331887	image188.png 0.331971

Table 43 BER results after clock wise 180° Rotation attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.250374	0.332038	0.249352
2	Standard Deviation	0.000077	0.000046	0.000021
3	Highest Value	image155.png 0.250583	image255.png 0.332182	image57.png 0.249410
4	Lowest Value	image24.png 0.250142	image151.png 0.331911	image135.png 0.249290

Tables 54, 55 and 56 does comparison on the basis of PSNR and BER with available techniques in literature. It can be seen in Table 56 that the algorithm is able to recover data from all the occlusion encrypted images.

From the results it can be seen that our proposed technique performs perfectly for $1/64$, $1/16$, $1/4$, $1/2$, $3/4$ occluded part in the encrypted image as shown in Table 49, 50, 51, 52 and 53.

5.6.5 Crypto analysis (CA)

The cipher text only attack is an attack which is used for cryptanalysis where it is assumed that the unauthorized user have access to the cipher text. The cryptography fails to resist the attack

Table 44 BER results after anti-clock wise 90° rotation attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128,3)	BER (256,256,3)	BER (512,512,3)
1	Average	0.332035	0.332029	0.332028
2	Standard Deviation	0.000091	0.000048	0.000024
3	Highest Value	image76.png 0.332228	image161.png 0.332174	image191.png 0.332094
4	Lowest Value	image63.png 0.331740	image64.png 0.331904	image213.png 0.331964

Table 45 PSNR results, comparison results for Rotation attack with techniques available in the literature on set-b

Serial Number	Technique Name	CWR-90°	CWR-180°	ACWR-90°
1	I. Shatheesh Sam et al. [40]	–	27.05446	–
2	X. Liao et al. [25]	–	–	–
3	A. Akhshani et al. [3]	–	26.23864	–
4	I. Sam et al. [39]	–	27.21036	–
5	M. François et al. [16]	–	27.1165	–
6	A. A. Abd El-Latif et al. [2]	–	27.39047	–
7	M. SaberiKamarposhti et al. [38]	–	–	–
8	R. Bansal et al. [8]	–	26.96607	–
9	G. Hanchinamani et al. [18]	–	27.05446	–
10	H. Liu et al. [26]	–	26.23864	–
11	F. Hu et al. [21]	–	–	–
12	G. Ye et al. [45]	–	27.2165	–
13	M. Kumari et al. [22]	–	26.92937	–
14	N. Zhou et al. [47]	–	27.12226	–
15	X. Liu et al. [27]	–	26.23864	–
16	X. Zhang et al. [46]	–	–	–
17	C. He et al. [20]	–	–	–
18	M. Kumari et al. [23]	–	27.446173	–
19	Y. Ding et al. [14]	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–
21	Proposed Scheme	8.569810193	8.568771017	8.568629918

Table 46 BER results after vertical then rotation attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Technique Name	CWR-90°	CWR-180°	ACWR-90°
1	I. Shatheesh Sam et al. [40]	–	0.996257	–
2	X. Liao et al. [25]	–	–	–
3	A. Akhshani et al. [3]	–	0.992594	–
4	I. Sam et al. [39]	–	0.996582	–
5	M. François et al. [16]	–	0.996338	–
6	A. A. Abd El-Latif et al. [2]	–	0.994954	–
7	M. SaberiKamarposhti et al. [38]	–	–	–
8	R. Bansal et al. [8]	–	0.996017	–
9	G. Hanchinamani et al. [18]	–	0.996257	–
10	H. Liu et al. [26]	–	0.992594	–
11	F. Hu et al. [21]	–	–	–
12	G. Ye et al. [45]	–	0.996338	–
13	M. Kumari et al. [22]	–	0.997314	–
14	N. Zhou et al. [47]	–	0.996826	–
15	X. Liu et al. [27]	–	0.992594	–
16	X. Zhang et al. [46]	–	–	–
17	C. He et al. [20]	–	–	–
18	M. Kumari et al. [23]	–	0.994962	–
19	Y. Ding et al. [14]	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–
21	Proposed Scheme	0.332032	0.332038	0.332029

Table 47 Various plots generated for encrypted image corrupted by rotation attack on set-c

Plots For Set-C, Image Shape (512, 512, 3)		
<i>Original Input Image</i> 	<i>Encrypted Image</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Clock Wise Rotation 90 Attack Applied</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Clock Wise Rotation 180 Attack Applied</i> 	<i>Decrypted</i> 
<i>Original Input Image</i> 	<i>Clock Wise Rotation 270 Attack Applied</i> 	<i>Decrypted</i> 

Table 48 Different types of occluded part images in encrypted image i.e. $\frac{1}{64}, \frac{1}{16}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$


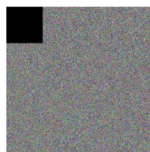
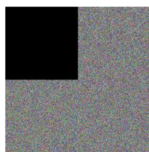
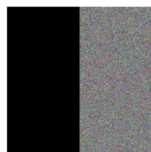

$\frac{1}{64}$	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$
				

Table 49 BER results after $1/64$ occlusion attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128)	BER (256,256)	BER (512,512)
1	Average	0.021764	0.008491	0.020998
2	Standard Deviation	0.000032	0.000008	0.000007
3	Highest Value	image104.png 0.021837	image178.png 0.008511	image46.png 0.021020
4	Lowest Value	image263.png 0.021640	image188.png 0.008465	image248.png 0.020979

Table 50 BER results after $1/16$ occlusion attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128)	BER (256,256)	BER (512,512)
1	Average	0.084990	0.032567	0.083497
2	Standard Deviation	0.000059	0.000015	0.000015
3	Highest Value	image171.png 0.085171	image264.png 0.032618	image18.png 0.083545
4	Lowest Value	image140.png 0.084791	image100.png 0.032513	image134.png 0.083455

Table 51 BER results after $1/4$ occlusion attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128)	BER (256,256)	BER (512,512)
1	Average	0.330712	0.126777	0.331706
2	Standard Deviation	0.000108	0.000029	0.000028
3	Highest Value	image181.png 0.331123	image151.png 0.126865	image223.png 0.331787
4	Lowest Value	image255.png 0.330390	image34.png 0.126663	image118.png 0.331629

Table 52 BER results after $1/2$ occlusion attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128)	BER (256,256)	BER (512,512)
1	Average	0.332007	0.167328	0.332031
2	Standard Deviation	0.000089	0.000034	0.000025
3	Highest Value	image82.png 0.332255	image71.png 0.167406	image122.png 0.332112
4	Lowest Value	image181.png 0.331740	image226.png 0.167223	image142.png 0.331968

Table 53 BER results after $3/4$ occlusion attack for the proposed algorithm on set-a, set-b and set-c respectively

Serial Number	Parameter	BER (128,128)	BER (256,256)	BER (512,512)
1	Average	0.332025	0.292759	0.332035
2	Standard Deviation	0.000088	0.000047	0.000024
3	Highest Value	image117.png 0.332275	image223.png 0.292886	image261.png 0.332105
4	Lowest Value	image74.png 0.331651	image137.png 0.292618	image120.png 0.331974

if the plaintext or key is obtained. [14, 31]. Table 57 depicts the cryptanalysis results of the of different techniques available literature on Chosen Plaintext Attack and Known plaintext Attack. Although out of these attacks, the chosen plaintext attack is the most threatening. As crypto analysis is the analysis and deciphering of cryptographic schemes, the cipher text attack is used for cryptanalysis, here it is assumed that the hacker have access to the cipher text. If the hacker does not get the key or plain text it is assumed that the cryptography passed to resist the

Table 54 PSNR results, comparison results for $1/64, 1/16, 1/4, 1/2, 3/4$ occlusion attack with techniques available in the literature on set-b

Serial Number	Technique Name	$1/64$	$1/16$	$1/4$	$1/2$	$3/4$
1	I. Shatheesh Sam et al. [40]	27.34669	27.34669	27.07675	26.72492	–
2	X. Liao et al. [25]	–	–	–	–	–
3	A. Akhshani et al. [3]	28.38841	27.98624	26.71644	25.87034	–
4	I. Sam et al. [39]	27.403	27.32936	27.04509	26.70974	–
5	M. François et al. [16]	27.34424	27.12346	26.34368	25.57719	–
6	A. A. Abd El-Latif et al. [2]	27.44282	27.42148	27.07829	26.72394	–
7	M. SaberiKamarposhti et al. [38]	–	–	–	–	–
8	R. Bansal et al. [8]	46.4718	40.9907	36.0379	34.676	–
9	G. Hanchinamani et al. [18]	25.61041	26.38365	26.38365	26.38365	–
10	H. Liu et al. [26]	32.1206	32.04338	31.77566	31.4934	–
11	F. Hu et al. [21]	–	–	–	–	–
12	G. Ye et al. [45]	27.49934	27.16829	26.40036	25.56192	–
13	M. Kumari et al. [22]	36.5365	34.2189	32.2707	32.106	–
14	N. Zhou et al. [47]	27.49237	27.44846	27.31067	27.25289	–
15	X. Liu et al. [27]	27.35341	27.16966	26.31166	25.56502	–
16	X. Zhang et al. [46]	–	–	–	–	–
17	C. He et al. [20]	–	–	–	–	–
18	M. Kumari et al. [23]	39.27967	34.24489	30.49161	29.66056	–
19	Y. Ding et al. [14]	–	–	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–	–	–
21	Proposed Scheme	24.48974409	18.64818484	12.75078473	11.54601708	9.116114123

Table 55 BER results, comparison results for $1/64, 1/16, 1/4, 1/2, 3/4$ occlusion attack with techniques available in the literature on set-b

Serial Number	Technique Name	$1/64$	$1/16$	$1/4$	$1/2$	$3/4$
1	I. Shatheesh Sam et al. [40]	0.996012	0.996175	0.996175	0.996012	–
2	X. Liao et al. [25]	–	–	–	–	–
3	A. Akhshani et al. [3]	0.996216	0.996216	0.997559	0.99585	–
4	I. Sam et al. [39]	0.993042	0.993103	0.993368	0.993835	–
5	M. François et al. [16]	0.995667	0.996277	0.996826	0.996643	–
6	A. A. Abd El-Latif et al. [2]	0.995687	0.995768	0.997152	0.995931	–
7	M. SaberiKamarposhti et al. [38]	–	–	–	–	–
8	R. Bansal et al. [8]	0.0142	0.0519	0.1662	0.2338	–
9	G. Hanchinamani et al. [18]	0.996094	0.995117	0.994873	0.995361	–
10	H. Liu et al. [26]	0.996419	0.996338	0.995931	0.994141	–
11	F. Hu et al. [21]	–	–	–	–	–
12	G. Ye et al. [45]	0.996094	0.996094	0.99585	0.997803	–
13	M. Kumari et al. [22]	0.7291	0.7538	0.7879	0.7907	–
14	N. Zhou et al. [47]	0.995361	0.995361	0.99528	0.995361	–
15	X. Liu et al. [27]	0.993896	0.993652	0.996582	0.996094	–
16	X. Zhang et al. [46]	–	–	–	–	–
17	C. He et al. [20]	–	–	–	–	–
18	M. Kumari et al. [23]	0.10074870	0.3198242	0.70768229	0.8147786	–
19	Y. Ding et al. [14]	–	–	–	–	–
20	M. Alkhelaiwi et al. [5]	–	–	–	–	–
21	Proposed Scheme	0.008491	0.032567	0.126777	0.167328	0.292759

attack. The other common cryptanalysis attack is known plaintext attack, where the hacker user has plaintext and the encrypted text to discover the key or text. The results show the proposed image encryption scheme is capable of resisting both the attacks.

5.7 Randomness test analysis (RTA)

This analysis helps to get an idea of the random nature of a PRNG. The results are computed using the NIST SP 800–22, Chi-Square Test and Runs Test on a highly randomized series generated by the GAN based PRNG. The results of the following test are shown in Tables 58, 59 and 60 respectively.

5.7.1 NIST SP 800–22

NIST SP 800–22A [19, 37] is basically a statistical test suite that is used for the analysis of random and pseudorandom number generators. It checks for the randomness of the algorithm. The proposed GAN based PRNG passes the NIST SP 800–22 test as shown in Table 58, from the results we can deduce that the training results of low-dimensional chaotic system are capable of passing the NIST test when the chaotic sequence is used as the training set.

Table 56 Various plots generated for encrypted image corrupted by $1/64, 1/16, 1/4, 1/2, 3/4$ occlusion attack on set-b


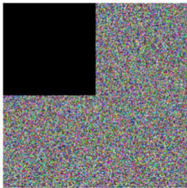

Plots For Set-B, Image Shape (256,256,3)		
<p>Original Input Image</p> 	<p>Anti-Occlusion Attack Applied</p> 	<p>Decrypted</p> 
<p>Original Input Image</p> 	<p>Anti-Occlusion Attack Applied</p> 	<p>Decrypted</p> 
<p>Original Input Image</p> 	<p>Anti-Occlusion Attack Applied</p> 	<p>Decrypted</p> 
<p>Original Input Image</p> 	<p>Anti-Occlusion Attack Applied</p> 	<p>Decrypted</p> 
<p>Original Input Image</p> 	<p>Anti-Occlusion Attack Applied</p> 	<p>Decrypted</p> 

Table 57 Crypto analysis attack results for chosen plaintext attack and known plaintext attack with techniques available in the literature on set-b

Serial Number	Technique Name	CPA	KPA
1	I. Shatheesh Sam et al. [40]	Yes	Yes
2	X. Liao et al. [25]	Yes	Yes
3	A. Akhshani et al. [3]	Yes	Yes
4	I. Sam et al. [39]	Yes	Yes
5	M. François et al. [16]	Yes	Yes
6	A. A. Abd El-Latif et al. [2]	Yes	Yes
7	M. SaberiKamarposhti et al. [38]	Yes	Yes
8	R. Bansal et al. [8]	Yes	Yes
9	G. Hanchinamani et al. [18]	Yes	Yes
10	H. Liu et al. [26]	Yes	Yes
11	F. Hu et al. [21]	Yes	Yes
12	G. Ye et al. [45]	Yes	Yes
13	M. Kumari et al. [22]	Yes	Yes
14	N. Zhou et al. [47]	Yes	Yes
15	X. Liu et al. [27]	Yes	Yes
16	X. Zhang et al. [46]	Yes	Yes
17	C. He et al. [20]	Yes	Yes
18	M. Kumari et al. [23]	Yes	Yes
19	Y. Ding et al. [14]	Yes	Yes
20	M. Alkhelaiwi et al. [5]	Yes	Yes
21	Proposed Scheme	Yes	Yes

Table 58 NIST SP 800–22 test suit results for randomness test

Serial Number	Test name	P Value	Status
1	Frequency	0.7165	Success
2	Block Frequency	0.8861	Success
3	Cumulative Sums	0.9759	Success
4	Runs	0.3254	Success
5	Longest Run of Ones	0.5087	Success
6	Rank	0.3203	Success
7	Discrete Fourier Transform	0.1264	Success
8	Nonperiodic Template Matchings	0.6989	Success
9	Overlapping Template Matchings	0.2194	Success
10	Universal Statistical	0.0157	Success
11	Approximate Entropy	0.9781	Success
12	Random Excursions	0.5995	Success
13	Random Excursions Variant	0.7964	Success
14	Serial	0.8253	Success
		0.7978	
15	Linear Complexity	0.6264	Success

Table 59 Chi-square test results for randomness test

Serial Number	Points	Statistic	P Value
1	10	0.5800290050759064	0.031497417586721255
2	100	1683.3333333333333	2.6665552686776317e-286
3	1000	166,833.3333333333	0.0
4	10,000	16,668,333.333333334	0.0
5	100,000	1,666,683,333.3333335	0.0
6	1,000,000	166,666,833,333.3333	0.0

5.7.2 Chi-square test (CST)

Chi-square test is basically used to check the regularity between the original image and the encrypted image. Lower is the chi-square value; the better is the consistency, resulting in superior degree of the encryption effect. [31] The proposed GAN based PRNG passes the Chi-Square test as shown in Table 59. This means the series generated by the proposed PRNG are highly randomized and exhibits highly random nature.

5.7.3 Runs test (RT)

A runs test is a statistical test, also known as the Wald–Wolfowitz runs test that examines if data is occurring randomly or not, where a run is defined as a series of increasing values or a series of decreasing values. The proposed GAN based PRNG passes the Runs test as $Z_{statistic} < Z_{critical}$ where the value of $Z_{statistic} = 0.5700002850004988$ and $Z_{critical} = 1.96$ as shown in Table 60. This means the series generated by the proposed PRNG are highly randomized and exhibits highly random nature.

6 Comparison with other state-of-the-art algorithms

The results in Table 61 show that the proposed scheme overpowers renowned mechanisms in terms of Visual Analysis, Statistical Analysis, Quantitative Analysis (H, SSIM, and BER), Differential Attack Analysis, Other Attack Analysis and Randomness Test Analysis.

Table 60 Runs test results for randomness test

Serial Number	Points	$Z_{statistic}$	Status
1	10	1.3416407864998738	Success
2	100	1.0050890913907349	Success
3	1000	1.328821859795413	Success
4	10,000	0.5800290050759064	Success
5	100,000	1.8910514961563638	Success
6	1,000,000	0.5700002850004988	Success

Table 61 Comparison table with other state-of-the-art algorithms

References	Year	VA	SA	QA			DAA			OAA			RTA				
				CC	HA	MSE	PSNR	H	SSIM	BER	NPCR	UACI	BFST	NA	GA	CA	AOA
F. Hu et al. [21]	2017	✓	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	×	✓
C. He et al. [20]	2019	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	×	✓
Y. Ding et al. [14]	2020	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	×	✓
M. Alkhalawi et al. [5]	2021	✓	×	✓	✓	✓	×	×	×	×	×	✓	×	✓	×	×	✓
Y. Cao et al. [11]	2022	✓	×	✓	×	×	×	×	✓	✓	×	✓	×	✓	×	×	✓
Proposed Scheme		✓	✓	✓	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 62 Motivation

Serial Number	Reference	Year	IP /DL
1	M. Abadi et al. [1]	2016	DL
2	M. De Bernardi et al.[13]	2018	DL
3	Y. Ding et al. [14]	2020	DL

7 Conclusion

In this paper, a highly secure cryptography mechanism is proposed that accomplishes protection by employing a series of specially designed substitution boxes, permutation boxes, and diffusion boxes. It takes encryption keys as input, generated from a Generative Adversarial Network (GAN), an unsupervised deep learning algorithm trained on the Logistic Maps. It can be concluded from the experimental results that the proposed image encryption scheme outperforms state-of-the-art methods with better performance in all aspects. Since the trained generator passed the NIST test suite, chi-square test, and runs test. Further, it also withstands most of the probable attacks available in the literature; the algorithm even offered the promising average of MSE, PSNR, BER, SSIM, H, CC, NPCR, and UACI. The proposed algorithm provides a very high amount of brute force search time due to using a deep learning algorithm and performing operations at bit-level and byte-level. It also meets essentially all of the conditions that a good image encryption algorithm entails, which are listed as follows:

- The proposed scheme employs GAN an unsupervised deep learning based robust image encryption technique.
- The projected algorithm shows very high key randomness.
- The proposed scheme offers a very high key space.
- The given scheme offers byte level operations at substitution stage, and both byte level and bit level operations at the permutation and diffusion stages.
- The scheme offers a very high key sensitive mechanism for the generation of random numbers.

8 Motivation, applications and future scope

This Research aims the development of a robust image encryption scheme based on an unsupervised deep learning algorithm and multilevel encryption technique, which can withstand against wide range of attacks. Table 62 shows the list of papers from which this research is motivated.

From the research conducted in this paper it can be deduced that, a deep neural network can be trained on less random sequences, and can be used as a highly randomized PRNG for creating unpredictable and extremely random sequences. These neural networks based PRNGs can be utilized for various applications such as image encryption mechanisms for securing image data, hybrid security algorithms to store text data at random locations in an image, gaming applications where highly randomized behaviour is required and generating random captchas for online authentication etc. With advancement in cybercrime it is also required to

search new ways to counter these advancements. Employment of machine learning and deep learning techniques can reduce this advancement. This opens the door for further research in the field of encryption to secure data.

Data availability The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interest The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

References

1. Abadi M, Andersen DG (2016) Learning to protect communications with adversarial neural cryptography. arXiv 2016. arXiv preprint arXiv:1610.06918
2. Abd El-Latif AA, Li L, Wang N, Han Q, Niu X (Nov. 2013) A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces. *Signal Process* 93(11):2986–3000. <https://doi.org/10.1016/j.sigpro.2013.03.031>
3. Akhshani A, Akhavan A, Lim S-C, Hassan Z (Dec. 2012) An image encryption scheme based on quantum logistic map. *Commun Nonlinear Sci Numer Simul* 17(12):4653–4661. <https://doi.org/10.1016/j.cnsns.2012.05.033>
4. Akhshani A, Akhavan Masoumi A, Mobaraki A, Siew-Choo L, Hassan Z (Jan. 2014) Pseudo random number generator based on quantum chaotic map. *Commun Nonlinear Sci Numer Simul* 19:101–111. <https://doi.org/10.1016/j.cnsns.2013.06.017>
5. Alkhalaiwi M, Boulila W, Ahmad J, Koubaa A, Driss M (2021) An Efficient Approach Based on Privacy-Preserving Deep Learning for Satellite Image Classification. *Remote Sens* 13(11):2221. <https://doi.org/10.3390/rs13112221>
6. Appel AW (Apr. 2015) Verification of a cryptographic primitive: SHA-256. *ACM Trans Program Lang Syst* 37(2):1–31. <https://doi.org/10.1145/2701415>
7. Arjovsky M, Chintala S, Bottou L (Aug. 2017) Wasserstein Generative Adversarial Networks. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 214–223. [Online]. Available: <http://proceedings.mlr.press/v70/arjovsky17a.html>. Accessed Aug 2022
8. Bansal R, Gupta S, Sharma G (Aug. 2017) An innovative image encryption scheme based on chaotic map and Vigenère scheme. *Multimed Tools Appl* 76(15):16529–16562. <https://doi.org/10.1007/s11042-016-3926-9>
9. Brock A, Donahue J, Simonyan K (2018) Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*
10. Brunet D, Vrscay ER, Wang Z (Apr. 2012) On the mathematical properties of the structural similarity index. *IEEE Trans Image Process* 21(4):1488–1499. <https://doi.org/10.1109/TIP.2011.2173206>
11. Cao Y, Shi P, Wu K, Li W (May 2022) Image encryption algorithm based on an improved ML neuron model and DNA dynamic coding. *Comput Intell Neurosci* 2022:4316163. <https://doi.org/10.1155/2022/4316163>
12. Chopra A, Gupta S, Dhall S (Jan. 2020) Analysis of frequency domain watermarking techniques in presence of geometric and simple attacks. *Multimed Tools Appl* 79(1):501–554. <https://doi.org/10.1007/s11042-019-08087-x>
13. De Bernardi M, Khouzani MHR, Malacaria P (2019) Pseudo-Random Number Generation Using Generative Adversarial Networks. In: *ECML PKDD 2018 Workshops*, Cham, pp. 191–200
14. Ding Y et al (2020) DeepEDN: a deep learning-based image encryption and decryption network for internet of medical things
15. Dosselmann R, Yang XD (Mar. 2011) A comprehensive assessment of the structural similarity index. *SIViP* 5(1):81–91. <https://doi.org/10.1007/s11760-009-0144-1>
16. François M, Grosset T, Barchiesi D, Erra R (2012) A new image encryption scheme based on a chaotic function. *Signal Process Image Commun* 27(3):249–259. <https://doi.org/10.1016/j.image.2011.11.003>
17. Goodfellow I et al (2014) Generative adversarial nets. *Adv Neural Inf Process Syst* 27

18. Hanchinamani G, Kulkarni L (Jul. 2015) An Efficient Image Encryption Scheme Based on a Peter De Jong Chaotic Map and a RC4 Stream Cipher. *3D Res* 6(3):30. <https://doi.org/10.1007/s13319-015-0062-7>
19. Hars L, Petruska G (Feb. 2012) Pseudorandom recursions II. *EURASIP J Embed Syst* 2012(1):1. <https://doi.org/10.1186/1687-3963-2012-1>
20. He C, Ming K, Wang Y, Wang ZJ (2019) A deep learning based attack for the Chaos-based image encryption
21. Hu F, Wang J, Xu X, Pu C, Peng T (Feb. 2017) Batch image encryption using generated deep features based on stacked autoencoder network. *Math Probl Eng* 2017:3675459. <https://doi.org/10.1155/2017/3675459>
22. Kumari M, Gupta S (Mar. 2018) A novel image encryption scheme based on intertwining chaotic maps and RC4 stream cipher. *3D Res* 9(1):10. <https://doi.org/10.1007/s13319-018-0162-2>
23. Kumari M, Gupta S, Malik A (Nov. 2020) A superlative image encryption technique based on bit plane using key-based electronic code book. *Multimed Tools Appl* 79(43):33161–33191. <https://doi.org/10.1007/s11042-020-09627-6>
24. Ledig C et al (Jul. 2017) Photo-realistic single image super-resolution using a generative adversarial network
25. Liao X, Lai S, Zhou Q (Sep. 2010) A novel image encryption algorithm based on self-adaptive wave transmission. *Signal Process* 90(9):2714–2722. <https://doi.org/10.1016/j.sigpro.2010.03.022>
26. Liu H, Jin C (Jan. 2017) A Novel Color Image Encryption Algorithm Based on Quantum Chaos Sequence. *3D Res* 8(1):4. <https://doi.org/10.1007/s13319-016-0114-7>
27. Liu X, Xiao D, Xiang Y (2019) Quantum image encryption using intra and inter bit permutation based on logistic map. *IEEE Access* 7:6937–6946. <https://doi.org/10.1109/ACCESS.2018.2889896>
28. Liu H, Xu Y, Ma C (2020) Chaos-based image hybrid encryption algorithm using key stretching and hash feedback. *Optik* 216:164925
29. Liu H, Kadir A, Chengbo X (2020) Color image encryption with cipher feedback and coupling chaotic map. *Int J Bifurcat Chaos* 30(12):2050173
30. Liu H, Liu J, Ma C (2022) Constructing dynamic strong S-Box using 3D chaotic map and application to image encryption. *Multimed Tools Appl*:1–16. <https://doi.org/10.1007/s11042-022-12069-x>
31. Malik A, Jadav S, Gupta S (Jun. 2021) Assessment of diverse image encryption mechanisms under prevalent invasion. *Multimed Tools Appl* 80(14):21521–21559. <https://doi.org/10.1007/s11042-021-10670-0>
32. Man Z et al (May 2021) A novel image encryption algorithm based on least squares generative adversarial network random number generator. *Multimed Tools Appl* 80:27445–27469. <https://doi.org/10.1007/s11042-021-10979-w>
33. Martino R, Cilaro A (Sep. 2020) Designing a SHA-256 processor for blockchain-based IoT applications. *Internet of Things* 11:100254. <https://doi.org/10.1016/j.iot.2020.100254>
34. Mendel F, Pramstaller N, Rechberger C, Rijmen V (2006) Analysis of Step-Reduced SHA-256. In: *Fast Software Encryption, Berlin, Heidelberg*, pp 126–143
35. Mirza M, Osindero S (2014) Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*
36. Padi M, Chaudhari R (Dec. 2017) An optimized pipelined architecture of SHA-256 hash function. In: *2017 7th International Symposium on Embedded Computing and System Design (ISED)*, pp. 1–4. <https://doi.org/10.1109/ISED.2017.8303943>
37. Rukhin A et al (2010) NIST special publication 800-22: a statistical test suite for the validation of random number generators and Pseudo random number generators for cryptographic applications,” NIST Special Publication 800-22
38. SaberiKamarposhti M, Mohammad D, Shafry Mohd Rahim M, Yaghoobi M (Feb. 2014) Using 3-cell chaotic map for image encryption based on biological operations. *Nonlinear Dyn* 75(3):407–416. <https://doi.org/10.1007/s11071-013-0819-6>
39. Sam I, Ponnaian D, Bhuvaneshwaran RS (Nov. 2012) A novel image cipher based on mixed transformed logistic maps. *Multimed Tools Appl* 56:315–330. <https://doi.org/10.1007/s11042-010-0652-6>
40. Shatheesh Sam I, Devaraj P, Bhuvaneshwaran RS (Sep. 2012) An intertwining chaotic maps based image encryption scheme. *Nonlinear Dyn* 69(4):1995–2007. <https://doi.org/10.1007/s11071-012-0402-6>
41. Wang Z, Bovik AC (Mar. 2002) A universal image quality index. *IEEE Signal Process Lett* 9(3):81–84. <https://doi.org/10.1109/97.995823>
42. Wang Z, Bovik AC (Jan. 2009) Mean squared error: love it or leave it? A new look at signal Fidelity measures. *IEEE Signal Process Mag* 26(1):98–117. <https://doi.org/10.1109/MSP.2008.930649>
43. Wang J, Hu Y (2020) An improved enhancement algorithm based on CNN applicable for weak contrast images. *IEEE Access* 8:8459–8476. <https://doi.org/10.1109/ACCESS.2019.2963478>
44. Ye G, Wong K-W (Sep. 2012) An efficient chaotic image encryption algorithm based on a generalized Arnold map. *Nonlinear Dyn* 69(4):2079–2087. <https://doi.org/10.1007/s11071-012-0409-z>

45. Ye G, Pan C, Huang X, Zhao Z, He J (2018) A chaotic image encryption algorithm based on information entropy. *Int J Bifurcat Chaos* 28(01):1850010. <https://doi.org/10.1142/S0218127418500104>
46. Zhang X, Wang L, Cui G, Niu Y (Aug. 2019) Entropy-based block scrambling image encryption using DES structure and chaotic systems. *Int J Optics* 2019:3594534. <https://doi.org/10.1155/2019/3594534>
47. Zhou N, Chen W, Yan X, Wang Y (Apr. 2018) Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system. *Quantum Inf Process* 17(6):137. <https://doi.org/10.1007/s11128-018-1902-1>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.