RESEARCH ARTICLE OPEN ACCESS

Securing Colour Image Based on DNA Encoding and Chaos

Tayseer I Salman [1], Kinda S Abou Kassem [2]
Computer Engineering and Automatic Control
Faculty of Mechanical and Electrical Engineering, Tishreen University
Latakia, Syria

ABSTRACT

The objective of this study is to provide a method for encrypting colour images and to enhance key space. It makes use of DNA encoding and DNA complement operation. Using SHA-512 that depends on plain image to produce a key to update 3D Lorentz system which in turn produces three random sequences to transform pixel locations. This paper uses an external DNA secret key with 512 symbol length to make better protection from brute force attacks. Moreover the resulted ciphered image is kept in DNA codes before transmitting or storing which enhances security. The proposed method passed the required security measures such as correlation of adjacent pixels, key space, key sensitivity, entropy, histogram, the number of pixel change rate (NPCR) and unified averaged changed intensity (UACI).

Keywords:- DNA, chaos, key space, key sensitivity, information entropy, NPCR, UACI.

I. INTRODUCTION

The main idea of image encryption is to change the values and pixel positions of the plain image matrix and produces the cipher image. Traditional encryption methods adopted performing operations on these values for the purpose of confusing plain image using random transformations to change pixel positions and obtaining an image which is entirely different from the original image. Some traditional algorithms like DES, AES and IDEA for information security but cannot be applied directly to encrypt images and give satisfactory results because of some proprieties related to images such as big data size, high correlation between adjacent pixels and high redundancy [1].

A. Representing grey scale images

Previous studies [3]-[10] used DNA encoding in image encryption where the plain image is usually represented in a form of matrix $n * m$; In other words n is the number of columns and m is the number of rows in the image matrix. The pixel values for a grey scale image ranges from 0 to 255, then the pixel has one decimal value in the range of integer values [0, 255] where value 0 refers to black colour and 255 refers to white colour Fig. 1. In the case of coloured image, it represented using three matrices and every image element consists of three values or pixels.

Colour image in turn could be represented using three matrices as illustrated in Fig. 2. Every matrix of them refers to the amount of

Colour in the image thus we have red R, green G and blue B matrices respectively, all of these matrices are of greyscale colour in the range [0 – 255] is called RGB3. Elements of these matrices are integers between 0 and 255 which in turn determined pixel intensity of the colour in the matrix. This gives $256^3 = 2^{24} = 16777216$ different colour in the RGB system.

II. LITERATURE REVIEW

Many studies used chaos transforms to transform plain image into confused image as much as possible to reduce correlation coefficient between adjacent pixels. Some studies relayed on merging chaos maps such as Arnold Cat Map and Lorentz and DNA encoding to introduce additional protection layer.

Authors of paper [2] used Lorentz, Chen and Lu for image confusion on two stages. They used two keys with length 16 bytes, first key to confuse the image and the second key to change pixel values and based on the initial values of the key one of the three chaos maps Lorenz or Chen or Lu is selected and this map in turn confused the image as first stage, in the second stage, second key is used which is independent of the first key. Based on the initial values of the second key one of

the three maps is selected to change pixel values of the plain image.

The authors in [3] used 1D chaos maps, they combined two chaotic maps in parallel to produce ciphered image with high confusion and diffusion. The algorithm is of four rounds and each round takes five steps include random pixel insertion to the beginning of each row in the image matrix then row separation and permute values in each row, combine rows and rotate 90 degrees counter clock wise and eventually the encrypted image. The key space for this algorithm is 10^{84}.

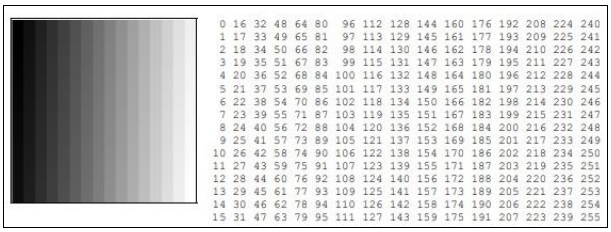

Fig. 1 pixel values for grey - scale image 16x16 with integers [4]

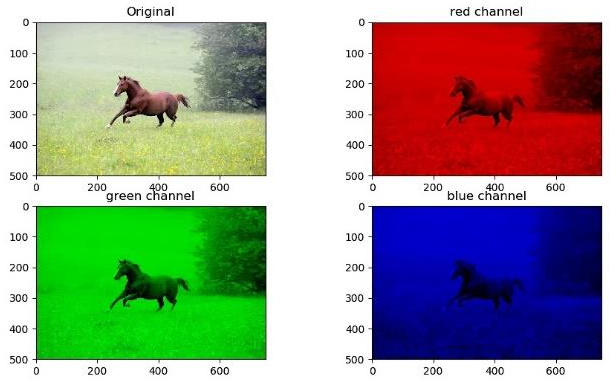

Fig. 2 representing coloured image with three colour channels RGB

The spatiotemporal chaos and DNA were deployed in [5] for image encryption. Image is diffused using XOR then DNA encoding again the DNA encoded image is diffused using the chaos sequence resulted from spatiotemporal chaos to produce encrypted image. This method is better suited for greyscale images. Key space obtained is 10^{93}.

The chaos maps of 1D and DNA operations have been deployed in [6]. Firstly Key generation is done using three 1D chaos maps depending on secret key and plain image. Random transformation for keys and plain image into DNA matrices using DNA encoding rules. Secondly performing DNA complement and XOR operation on the DNA matrices to produce scrambled DNA matrices then decomposing scrambled matrices into blocks then mix resulted blocks randomly. Eventually, apply DNA operations such as addition and XOR on matrices obtained from previous step then transforming DNA encoded matrices into encrypted image. This study gave good results in comparison to previous algorithms related to security measures and produced a key space of 2^{299}

The authors in [7] introduced a method to encrypt colour image using DNA encoding and hyper chaos. Encryption process is done in two stages. In the first stage, decomposition of coloured image into three matrices representing RGB colour components of the image then pixel to DNA encoding for each matrix. The scrambling process for the three matrices is done using Chen chaos function. In the second stage generating three chaos sequences using Logistic Map with external key of length 128 bit, all of these sequences are encoded using DNA. Encryption process is done using XOR between scrambled colour matrices and matrices resulted from chaos sequences and produces DNA encrypted image which in turn DNA decoded to get the encrypted image. Key space is $10^{70} \approx 2^{233}$.

The authors in [8] evaluated algorithms ZGW2013 and ZGW2012 and determined points of weakness and concluded that DNA encoding rules must be included in encryption key and features of the plain image must be extracted and used as parameter in generating random sequences and enhance the relation between DNA computing, permutation and confusion for images. They concluded that DNA computing cannot be applied directly in random encryption operations and permutation and confusion. Key space 10^{56}

A. Literature review summary

According to previously mentioned studies, image encryption process starts with reading the intended image which translated into $n * m$ matrix which every element in it representing pixel value. For grey scale images there is one value for pixel in the image which lays in the range [0-255], but in the case of colour image every image element consists of three colour values expressed with three pixel values. To encrypt an image either coloured or grey scale all literature refers to make confusion and diffusion for image matrix to reduce correlation coefficient between adjacent pixels as much as possible and make it near zero. Also the encryption tries to increase ciphers image's entropy or increasing the randomness of pixel distribution among the encrypted image matrix where the ideal value for entropy is 8 [7], [8]. To make confusion, pixel positions of the plain image are changed or transformed with keeping indexes to get back when decrypting. Diffusion is performed using a sufficient long secret key. Chaos logistic maps were used [1]-[3] to perform confusion and diffusion. Chaos and DNA encoding were adopted in [4]-[7] but none of these studies used

encryption keys from DNA sequences either true DNA or randomly generated DNA sequences.

III. PROPOSED METHOD

The proposed method is dedicated for colour image and makes use of DNA sequences to represent image with DNA encoding and makes use of chaos resulted from Lorentz system to change pixel positions for the plain image. The encryption process explained in the steps below which illustrated in Fig. 3:

1- Read plain colored image and convert it into $n * m$ pixel matrix.
2- Decompose original color image into three color component matrices R,G,B
3- Constructing key matrix which is of length $n * m * 4$, depending on plain image dimensions and a random DNA sequence with length $n * m * 4$ this DNA sequence taken either from general genetic database [9] or randomly generated.
4- Generate SHA-512 key from plain image.
5- Encode the R, G, B matrices with DNA codes by using one of the 8 encoding rules, Table. 1
6- Apply DNA complement rule on the three encoded matrices.
7- Updating initial values of Lorentz system x_0, y_0, z_0 based on the SHA-512 key.
8- Generate random sequences x, y, z from Lorentz map where each of x, y, z is 1D matrix with size $n * m * 4$ where (4 is the number of DNA symbols).
9- Perform indexing for the random sequences resulted from step 6 and get indexes fx, fy, fz.
10- Encrypt the three color matrices R, G, B with DNA Key matrix using XOR operation.
11- Pixel transposition for the three color matrices using the resulted values from 7.
12- Compose R, G, B matrices and get DNA encrypted image.

To recover original image back, a reverse process for steps 12 to 1 is carried out. The flow chart for the encryption algorithm is illustrated in Fig.3.

A. DNA encoding of the plain image

Pixel values of the plain image are converted into DNA codes considering original image in RGB system. At the beginning plain image is read then transformed into $n * m$ matrix where n is the number of columns, and m is the number of rows, then plain image is decomposed into three colour components R, G, B

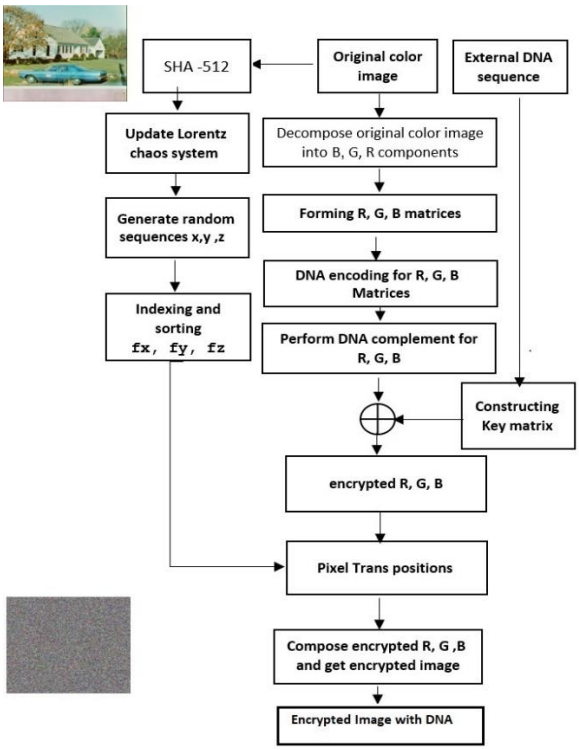

Fig. 3 encryption algorithm

each of them is a matrix with grey scale in the range [0-255] and each value in this range can be expressed in a series of four DNA symbols. DNA sequences consist of four symbols {A, C, G, T} that come in random sequence then there are $4^4 = 256$ pattern of four symbols which is suitable for representing image where pixel values lay in the range [0 - 255] then it's possible to change each value with the corresponding DNA pattern. For example pixel 0 maps to AAAA, pixel value 112 maps to CTAA, pixel value 255 maps to TTTT DNA coding respectively.

After obtaining original image matrix in the form of pixel value, the plain image is decomposed into three colour components R, G, B; these components are matrices with $m * n$ dimensions then DNA encode R, G, B matrices using one of the 8 encoding rules in Table. 1. The resulting matrices dimensions are $m * n * 4$ where m is the number of rows and 4 is the number of DNA symbols used for encoding each pixel value illustrated in Fig. 4

(a) original grey scale image 100X100

148	175	133 ...	215	221	127
79	150	127 ...	218	223	128
...	...	...	...	...	...
170	209	208 ...	165	167	121
172	212	210 ...	177	161	128
136	174	170 ...	149	132	103

(b) pixel values

GCCC	GGTT	GACC ...	TCCT	TCTA	CTTT
CATT	GCCG	CTTT ...	TCGG	TCTT	GAAA
...	...	...	...	...	...
GGGG	TCAG	TCAA ...	GGCC	GGCG	CTGC
GGTA	TCCA	TCAT ...	GTAC	GGAC	GAAA
GAGA	GGTG	GGGG ...	GCCA	GACA	CGCT

(c) DNA encoded image

Fig. 4 DNA encoding of original image

B. Complement operation on DNA encoded matrix

A complement operation is done after obtaining the three DNA encoded colour matrices. The complement operation is performed by converting A to T and G to C and vice versa C to G and T to A. Fig.5 illustrates complement operation. The purpose of complement operation is to change pixel values in the matrix and provides confusion for pixel values in three colour matrices. To change pixel positions in the colour matrices, chaos maps are used, in this paper 3D Lorentz chaos is used.

GCCC	GGTT	GACC ...	TCCT	TCTA	CTTT
CATT	GCCG	CTTT ...	TCGG	TCTT	GAAA
...	...	...	...	...	...
GGGG	TCAG	TCAA ...	GGCC	GGCG	CTGC
GGTA	TCCA	TCAT ...	GTAC	GGAC	GAAA
GAGA	GGTG	GGGG ...	GCCA	GACA	CGCT

(a)

CGGG	CCAA	CTGG ...	AGGA	AGAT	GAAA
GTAA	CGGC	GAAA ...	AGCC	AGAA	CTTT
...	...	...	...	...	...
CCCC	AGTC	AGTT ...	CCGG	CCGC	GACG
CCAT	AGGT	AGTA ...	CATG	CCTG	CTTT
CTGT	CCAC	CCCC ...	CGGT	CTGT	GCGA

(b)

148	175	133 ...	215	221	127
79	150	127 ...	218	223	128
...	...	...	...	...	...
170	209	208 ...	165	167	121
172	212	210 ...	177	161	128
136	174	170 ...	149	132	103

(a1)

106	80	122 ...	40	35	128
176	150	128 ...	37	32	127
...	...	...	...	...	...
85	45	47 ...	90	89	134
83	43	44 ...	78	94	127
123	81	80 ...	107	123	152

(b1)

Fig.5 image matrix DNA complement; (a) DNA encoded original image; (b) DNA encoded original image after complement operation; (a1) pixel values of the original image; (b1) pixel values after DNA complement operation

C. Key matrix construction

Key matrix has the same dimensions R, G, B of the original image and for the colour matrices R, G, B. A key of 512 bit length is generated from SHA-512 by taking original image as input for the hash function. This key will be used later to update Lorentz system to produce three chaotic sequences. Using the hash function is to guarantee that any change in original image will result in a change in the ciphered image [10] which in turn enhances the protection against statistical attacks. The resulted hash key is encoded into DNA using one of the rules in Table 1. Key matrix as shown in Fig. 6 is filled with DNA encoded values of the hash key, then the key matrix is ready to be used to diffuse the three colour matrices using XOR operation.

TGAG	TATG	ATGG	TGCG	CCCG	GCCC	...	ATCA	GAGC
TGTG	CTAT	AACA	GCTG	GCGT	TCCG	...	CGGG	TTCA
TACG	ACAC	CCTC	TGGC	GACG	GTAT	...	CACG	GGTC
CTTG	GAGT	AGCC	TCCA	TTAA	CGCA	...	CAAT	CGAT
TACA	GTGC	TCCC	TTAC	GACC	GATT	...	TATG	TGCG
ATGA	AGTC	GGTT	TACT	GTTG	CGTC	...	CGCA	CTGT
GGAG	TCTG	GAGG	GAGC	CCGT	GACT	...	TGAA	ATAT
....	....	....	....	....	....	...	....	...
ATCT	GGTG	TACC	TGGG	CATT	TCCT	...	GCTT	CAGG

Fig. 6 key matrix

D. Lorentz chaotic System

This paper uses 3D Lorentz chaotic System since it produces three chaotic sequences each of them is used to change pixel positions in every colour matrix R, G, and B. Lorentz system biases via initial conditions. Lorentz system expressed through three differential equations:

$$\frac{dx_1}{dt} = \alpha * (y_1 - x_1) \quad (1)$$

$$\frac{dy_1}{dt} = x_1 * (\beta - z_1) - y_1 \quad (2)$$

$$\frac{dz_1}{dt} = (x_1 * y_1) - (\mu * z_1) \quad (3)$$

Where $\alpha, \beta, and\ \mu$ are control parameters and x_1, y_1, z_1 are the initial values. Lorentz system became chaotic at the values $\alpha = 10, \beta = 28,\ \mu = 8/3$ and $x_1 = 1, y_1 = 1, z_1 = 1$ and produces three different chaotic sequences x, y, z. Fig. 7 shows the behaviour of the 3D chaotic Lorentz system.

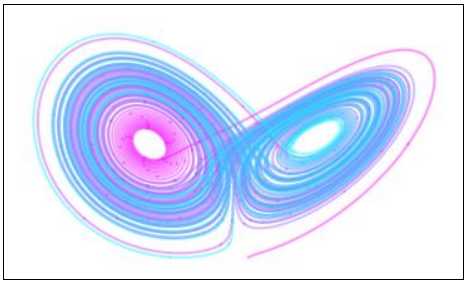

Fig .7 3D chaotic Lorentz system

The three chaotic sequences produced by Lorentz system are given in the following equations:

$$X = [x_i, x_{i+1}, x_{i+2}, \dots, x_{m*n*4-1}] \quad (4)$$
$$Y = [y_i, y_{i+1}, y_{i+2}, \dots, y_{m*n*4-1}] \quad (5)$$
$$Z = [z_i, z_{i+1}, z_{i+2}, \dots, z_{m*n*4-1}] \quad (6)$$

The resulted chaotic sequences are quantized to get three keys:

$$Lorentz\_Key_X = floor(mod(X\text{^}14,256)) \quad (7)$$
$$Lorentz\_Key_Y = floor(mod(Y\text{^}14,256)) \quad (8)$$
$$Lorentz\_Key_Z = floor(mod(Z\text{^}14,256)) \quad (9)$$

Where $Lorentz\_Key_X, Lorentz\_Key_y, Lorentz\_Key_z$ are three keys from the three chaotic sequences. All of these keys are 1D arrays of size $n * m * 4$, Fig.8 sorting each of them and obtaining indexes, these indexes are used to change pixel positions by replacing each pixel position according to indexes of the sorted chaotic sequences.

X	index
0.226562512	1
0.250733478	2
0.203261741	3
0.235596203	4
0.838623658	5
0.291651113	6
0.244678568	7
0.217706023	8
-0.241770602	9

(a)

Sorted (X)	Index
-0.241770602	9
0.203261741	3
0.217706023	8
0.226562512	1
0.235596203	4
0.244678568	7
0.250733478	2
0.291651113	6
0.838623658	5

(b)

73	33	92
57	35	40
60	80	53

(e) Confused Image

57	35	40
60	53	92
33	80	73

(d) Plain Image

9	3	8
1	4	7
2	6	5

(c) Index matrix

Fig .8 (a) sequence X; (b) sorted sequence X; (c) indexes; (d) pixel values for original image; (e) confused image using indexes

IV. WORK ENVIRONMENT

Implementation of the proposed algorithm is done using Python 3.7 with library OpenCv and hardware with GHz 3.5 Intel Core i7, 8GB RAM Windows 10 pro.

TABLE 1
DNA ENCODING RULES

	1	2	3	4	5	6	7	8
00	A	A	C	G	C	G	T	**T**
01	C	G	A	A	T	T	C	**G**
10	G	C	T	T	A	A	G	**C**
11	T	T	G	C	G	C	A	**A**

V. Security analysis

Security analysis of image encryption algorithm depends on several parameters:

A. Key sensitivity:

Key sensitivity means a tiny change in encryption key will lead to big change in the resulted ciphered image In another words, making a little change in the key then trying to decrypt the encrypted image then decryption process will fail and will not be able to recover the encrypted image. Fig. 9 shows key sensitivity when key changed by 10^{-6} then encrypted image couldn't be recovered.

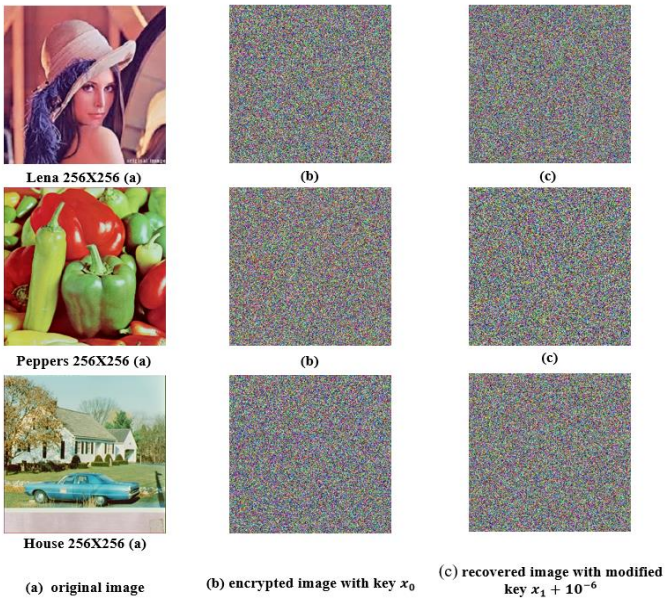

Fig. 9 key sensitivity

A. Key space

Key space is the number of keys to be tried to recover the secret key. As Kirchhoff stated [11] that *the encryption system should be safe even everything is known about it except the key*. To have a secure encryption algorithm, key space must be around 2^{100} *bit* size [5]. In this paper, the chaos system has three initial variables and precision of 10^{-14}. To calculate key space,

we have three sequences produced by the Lorentz system then they key space resulted from chaos is $10^{14} \times 10^{14} \times 10^{14} = 10^{42}$ in addition to the external DNA key with size 512 symbol and each symbol is encoded with 2bit then we have the total key space $(2^{1024})(10^{42}) = 2^{1163}$ which is large enough to resist brute force attack. In Table. 2 comparison of key space for the proposed and other references.

TABLE 2
KEY SPACE COMPARISON

Reference	Key space
[3]	$10^{84} = 2^{279}$
[4]	$10^{180} = 2^{600}$
[5]	$10^{93} = 2^{308}$
[6]	$10^{90} = 2^{299}$
[7]	$10^{70} = 2^{233}$
[8]	$10^{56} = 2^{186}$
Proposed	$10^{350} = 2^{1163}$

B. Correlation Analysis

Correlation analysis studies the relation between adjacent pixels in grey scale images. The more weak correlation in encrypted images the more efficient is the encryption algorithm is [12]. The correlation coefficient between adjacent pixels is calculated using three directions, Horisontal, Vertical, and Diagonal. A set of random pixels in the encerypted image is taken, a number of 3500 for example. The correlation coefficient between two adjacent pixels x, y denoted as $C_{r(x,y)}$ is carried out using the following equations:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (4)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad (5)$$

$$C_{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \quad (6)$$

$$C_{r(x,y)} = \frac{C_{cov}(x,y)}{\sqrt{D(x)\times D(y)}} \quad (7)$$

Where x, y are the grey scale of adjacent pixels and N is the total number of pixels chosen from image and $C_{cov}(x,y)$ is the covariance, $D(y), D(x)$ is the standard deviation and $E(x)$ is the average. Fig. 10 shows correlation for standard images (Baboon 512x512 and Lena 256x256) and Table 3 shows correlation coefficient comparison with other studies. Results taken for 15 rounds.

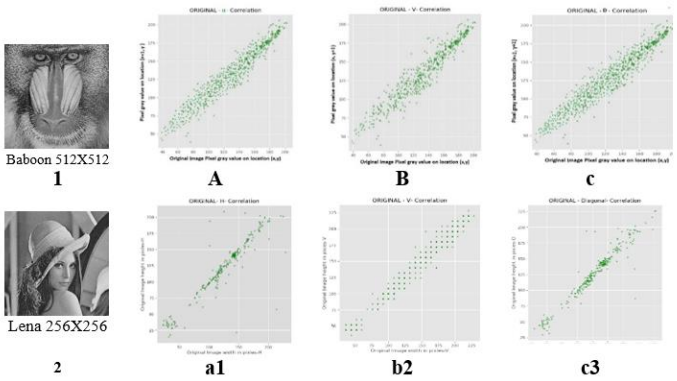

Fig. 10 (A, a1) Horizontal correlation- (x, y) pixel location on horizontal axis and pixel location (x+1, y) on vertical axis (B, b2) Vertical correlation- (x, y) pixel location on horizontal axis and pixel location (x, y+1) on vertical axis (c, c3) Diagonal correlation- (x, y) pixel location on horizontal axis and pixel location (x+1, y+1) on vertical axis for images 1 and 2 respectively.

TABLE 3
CORRELATION COEFFICIENT

Ref	Correlation coefficient		
	Encrypted Image		
	Horizontal	Vertical	**Diagonal**
Proposed	0.0012	0.0027	**0.0019**
[12]	-0.0021	-0.0032	**0.0037**
[13]	0.1257	0.0581	**0.0504**
[14]	0.0681	0.0845	-
[15]	0.0044	0.0034	**0.0020**
[16]	0.0024	0.0580	**0.0170**

E. Information Entropy

Information entropy is being used to evaluate grey pixel distribution in the original or encrypted image. The higher entropy rate the more chaotic the image. In other words the more uniform distribution the more encryption is able to resist statistical attacks. The ideal entropy value is 8 for grey scale image and truly random encrypted. Table 4 shows entropy for standard image Lena 255x255. Entropy is calculated with the following equation:

$$H(X) = -\sum_{i=1}^{L} P(x_i) log_2 \mathrm{P}(x_i) \quad (8)$$

$$\mathrm{P}(X = x_i) = \frac{1}{F} \quad (9)$$

Where x_i is the greyscale value and $\mathrm{P}(x_i)$ is the grey level probability for (x_i)

TABLE. 4
Plain Image Lena 255X255 entropy 7.4139

Image /Reference	Entropy rate
[Proposed]	**7.9996**
[17]	**7.9994**
[18]	**7.9970**
[19]	**7.9980**
[20]	**7.9975**

F. Histogram analysis

Histogram analysis Fig. 11 gives an idea about encryption efficiency against statistical attacks. If the histogram of uniform distribution for grey image pixels which shows a flat histogram. Encrypted images with flat histogram can resist statistical attack effectively [21]. Variances of the histogram are given in equation (10):

$$var(X) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)^2 \quad (10)$$

$X = \{x_0, x_1, x_2, \ldots . x_{255}\}$ Represents the histogram values vector and x_i, x_j are grey scale pixels which grey values are $i\ and\ j$ respectively and n represents grey level. When variance values are small then distribution is uniform and this is suitable to resist statistical attacks [22].

G. *Differential attacks*

In this type of attacks, the attacker randomly choses, a number of original images then encrypt them using the encryption algorithm in course and obtains the corresponding encrypted images. Differential analysis pairs of original, encrypted images and performs comparisons, in worst case key could be retrieved. A secure encryption algorithm must be sensitive to the original image. If a tiny change in the original image (one pixel change) it must lead to a big change between the two ciphered images resulted from the same original image and one-pixel changed original image, this change lead to huge difference between the two resulted cipher images. This difference can be calculated by two factors using the following two formulas:

$$NPCR = \frac{\sum_u D(i,j)}{m \times n} \times 100\% \quad (11)$$

$$UACI = \frac{1}{m \times n}\left[\sum_{ij}\frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \quad (12)$$

Where $C_1$the ciphered image of the original image and C_2 is the ciphered image after changing grey value for one pixel. While m is the row size and n is the column size. $NPCR$ Refers to the relation between original image and encrypted image, if pixel values are equal and have same position in the two matrices expressing original and ciphered images, then $D(i,j) = 0$

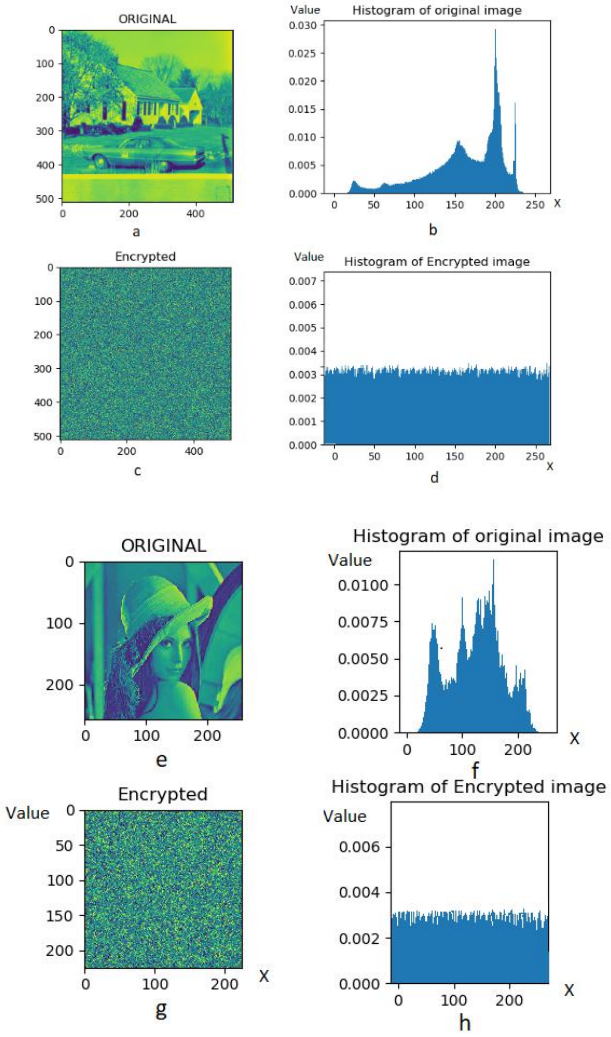

Fig. 11 (a) plain image – (b) plain image histogram – (c) encrypted image (d) encrypted image histogram – (e) plain image –(f) plain image histogram – (g) encrypted image (h) encrypted image

Else $D(i,j) = 1$, this is expressed as follows:

$$D(i,j) = \begin{cases} 0 & C(i,j) = \acute{C}(i,j) \\ 1 & C(i,j) \neq \acute{C}(i,j) \end{cases} \quad (13)$$

The ideal value for $NPCR$ is 99.61 [23]. And $UACI$ factor, refers to average intensity between two images. The ideal value for $UACI$ is 33.46 Table 6 shows the $NPCR$ and $UACI$ for the proposed and reference for the standard image Lena.

TABLE 6
NPCR AND UACI
for the standard image Lena 256*256

Ref	Proposed	[20]	[25]	[1]	**[24]**
NPCR	99.99	99.64	68.1731	99.5723	**99.6378**
UACI	33.76	33.63	31.7168	33.43	**33.6875**

An encryption scheme having $UACI$ greater than the theoretical value and $NPCR$ close to 100% becomes more effective regarding resisting differential attacks. To measure both values, a random change of one bit in the original image then repeat the process several times and take the average.

VI. Conclusions and discussion

An algorithm is introduced to encrypt colour images depending on external DNA key and original coloured image to generate initial values for Lorentz system via hash function SHA3 – 512. The external DNA key is of length 512 is used to make key matrix. A first step of pixel confusion is done by using DNA complement rule which adds an extra protection layer. DNA encoding is done for the three components of the image. Pixel transposition is done using chaotic Lorentz sequences and pixel diffusion is obtained by using the external random DNA sequence which contributed in making the image encryption more random. Moreover the resulted ciphered image is kept in DNA codes before transmitting or storing which enhances security. Using DNA key of length 512 and each DNA maps to $2\ bit$ which gives key of length $1024\ bit$ in addition to the three sequences produced by the chaotic system that give 10^{42} , this give a total key length $(2^{1024})(10^{42}) = 2^{1163}$ which is long enough to tackle brute force attacks. The proposed method passed all security tests with values as stated above.

REFERENCES

[1] T. Li, M. Yang, J. Wu and X. Jing, "A Novel Image Encryption Algorithm Based on a Fractional-Order Hyperchaotic System and DNA Computing," *Hindawi*, p. 13, 2017.

[2] K. Sakthidasan, Sankaran and B. V. S. Krishna, "A New Chaotic Algorithm for Image Encryption and Decryption of Digital Colour Images," *International Journal of Information and Education Technology*, vol. 1, no. 2, pp. 137 - 141, 2011.

[3] Y. Zhou, LongBao and C.L.PhilipChen, "A new1D chaotic system for image encryption," *Elsevier*, no. 97, pp. 172 - 182, 2013.

[4] V. Vučković, "IMAGE AND ITS MATRIX, MATRIX AND ITS IMAGE," *Преглед НЦД* , vol. 12, pp. 17 - 31, 2008.

[5] C. Song and Y. Qiao, "A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos," *Entropy*, no. 17, pp. 6954-6968, 2015.

[6] X. Wua, H. Kana and J. Kurths, "A new colour image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Elsevier*, pp. 1-16, 2015.

[7] A. Y. Niyat, R. M. H. Hei and M. V. Jahan, "A RGB image encryption algorithm based on DNA sequence operation and hyper-chaotic system," in *ICTCK 2015*, 2015.

[8] X.-Y. Wang, P. Li, Y.-Q. Zhang, L.-Y. Liu, H. Zhang and X. Wang, "A novel colour image encryption scheme using DNA permutation based on the Lorenz system," *Springer Science+Business Media New York*, pp. 1-23, 2017.

[9] Ncbi.nlm.nih.gov, "National Center for Biotechnology Information," NCBI, [Online]. Available: http://www.ncbi.nlm.nih.gov. [Accessed 15 11 2019].

[10] X. Wang, Y. Wang, X. Zhu and S. Unar, "Image encryption scheme based on Chaos and DNA plane operations," *Multimedia Tools and Applications*, pp. 1-18, 2019.

[11] K. principle, "crypto-it.net/eng," 2015. [Online]. Available: http://crypto-it.net/eng/theory/kerckhofs.html. [Accessed 2019].

[12] C. Song and Y. Qiao, "A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos," *Entropy*, no. 17, pp. 6954-6968, 2015.

[13] R. Rhouma, S. Meherzi and S. Belghith, "OCML-based colour image encryption," *Solitons Fractals*, vol. 40, no. 1, p. 309–318, 2009.

[14] B. Norouzi, S. Seyedzadeh, S. Mirzakuchaki and M. Mosavi, "A novel image encryption based on hash function with only two-round diffusion process," *Multimedia Syst*, vol. 20, no. 1, p. 45–64, 2013.

[15] R. Enayatifar, H. Sadaei, A. Abdullah, M. Lee and I. F. Isnin, "A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata," *Opt. Lasers Eng*, vol. 71, p. 33–41, 2015.

[16] H. Liu, X. Wang and A. Kadir, "Colour image encryption using Choquet fuzzy integral and hyper chaotic system," *Optik* , vol. 124, no. 14, p. 3527–3533, 2013.

[17] A. Jain and N. Rajpal, "A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps," *Multimed Tools Appl* , vol. 75, pp. 5455-5473 , 2016.

[18] S. Agarwal, "A Chaotic Cryptosystem using Conjugate Transcendental Fractal Function," *I. J. Computer Network and Information Security*, vol. 2, no. 1, pp. 1-12, 2019.

[19] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh and M.

R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimedia tools and Applications*, no. 71, p. 1469–1497, 2014.

[20] S. Stalin, P. Maheshwary, P. K. Shukla, M. Maheshwari, B. Gour and A. Khare, "Fast and Secure Medical Image Encryption Based on Non Linear 4D Logistic Map and DNA Sequences (NL4DLM_DNA)," *Journal of Medical Systems* , vol. 267, no. 43, pp. 1-17, 2019.

[21] C. T. Zhang, "Research on Image Encryption Based on DNA Sequence and Chaos Theory," in *Phys. Conf. Ser 1004 012023*, 2018.

[22] C. Xl, C. Yr and B. Lucie, "A novel chaos based image encryption algorithm using DNA sequence operations," *Opt Lasers Eng*, vol. 88, pp. 197 - 213, 2017.

[23] Y. Wang, K. Wong, X. Liao, T. Xiang and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," *Chaos Solitons Fractals*, vol. 41, no. 4, pp. 1773-83, 2009.

[24] X. LI, C. Zhou and N. Xu, "A Secure and Efficient Image Encryption Algorithm Based on DNA Coding and Spatiotemporal Chaos," *International Journal of Network Security*, vol. 20, no. 1, pp. 110-120, 2018.

[25] K. Zhan, D. Wei, J. Shi and J. Yu, "Cross-utilizing hyperchaotic and DNA sequences for image encryption," *Journal of Electronic Imaging, vol. 26, no. 1, Article ID 013021, 2017.*, vol. 26, no. 1, p. 13, 2017.