International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39324-39333

Image Encryption Algorithm Based on Graph Coloring

D. Sathya srinivas[1]
Research Scholar KarpagamUniversity Coimbatore

S. Veni[2]
Assistant Prof & Head Depart. of Computer Science Karpagam University, Coimbatore

ABSTRACT
Encryption is a method that ensures that the image transferred through the communication networks is protected against illegal reading, content alteration, addition of false information, or deletion of part of its content. Chaos based encryption techniques are considered good as these techniques provide a good combination of speed, high security, complexity, reasonable computational overheads and computational power. Chaotic sequences are real valued sequences. This real valued sequence can be converted into integer valued sequence. This generated sequence can be used for image encryption. Chaos-based image encryption schemes with control parameters are utilized to generate chaotic orbits employed to change the grey values of pixels so as to enhance the security. Recently, several chaotic based image encryption schemes have been proposed, with each of them has its own strengths and limitations in terms of security level and computational speed. In common, all chaotic encryption algorithms involve real number arithmetic calculations. In this paper, we propose a new image encryption algorithm based on graph coloring to increase the complexity of the encryption system so that it is very difficult to break it and predict on it. Our approach depends on generating a onetime pad with keys in random a manner to change the pixels value. The results of several experimental, statistical analysis, key sensitivity tests, NPCR and UACI analysis, and time analysis show that, the proposed image encryption algorithm provides an efficient and secure way for image encryption

Keywords: chaos, graph coloring, logistic map, henon map, baker map, tent map

1. INTRODUCTION

With the rapid developments in digital image processing and network communication, many applications such as military image databases, confidential video conference, telemedicine system, etc. communicate multimedia data over the Internet and wireless networks. However, when digital image is transmitted over the Internet, unauthorized individuals have the opportunities to access, copy, and destroy it. Hence, there is a need for a technique to secure the images from leakages, to provide a reliable, fast and robust secure system to store and transmit digital images and to protect digital image information against illegal copying and distribution. This has led to the development of image encryption algorithms. As image data capacity is very large and due to its intrinsic features like redundancy of data and strong correlation among adjacent pixels, traditional encryption algorithms such as DES, RSA are found to be not suitable for encrypting digital image. During the last decade, numerous encryption algorithms [1–6, 8-11, 16-21] have been proposed. Among them, chaos based encryption techniques are considered good. Chaotic systems have many important properties such as the sensitive dependence on initial conditions and system parameters, pseudorandom property, no periodicity and topological transitivity, etc. Most of these properties meet requirements such as diffusion and mixing in the sense of cryptography. Therefore, chaotic cryptosystems have been widely used in image encryption. Image encryption schemes on chaos involve time-consuming real number arithmetic operations. In this paper, we propose a new image encryption algorithm based on graph coloring to enhance the encryption system, increase the complexity of the encryption keys and decrease the computational complexity of the cipher image. First, the permutation process employs [22] to permute the image pixel positions totally. The proposed algorithm for diffusion generates aonetime pad which contains N random grey value sequences. The N sequences are used to modify the pixel grey values.

The organization of the rest of the paper is as follows: Section 2 gives a brief introduction of chaotic encryption algorithms and their limitations. Section 3and Section 4 present the proposed algorithm and the experimental results. Finally, Section 5 concludes the paper with a summary and a discussion of the future work.

2. RELATED WORK

A. Chaos Methods

The encryption algorithms such as Arnold transformation, magic square transformation, knight's tour transformation, Hilbert curve, Conway game, Fibonacci transformation, etc. have two disadvantages: one is high computational complexity, and the other is small key space. Chaos are used in the encryption system because of its characteristics like sensitivity to the initial conditions, high flexibility in the encryption system design, unpredictability, and good privacy due to both nonstandard approach and a vast number of variants of chaotic systems, large, complex and numerous possible encryption keys and simpler design. In addition to these characteristics, Chaos based encryption techniques provide a good combination of speed, high security, reasonable computational overheads, and computational power[**1**]. Chaos-based image encryption schemes[6] are usually composed of two processes: permutation process and diffusion process. Where the former permutes a plain-image with chaotic systems, the latter changes the pixel values sequentially so that a tiny change for one pixel can spread out to almost all pixels in the whole image. Among the existing

one-dimensional and two-dimensional chaotic systems, Logistic map, 2D tent map, Arnold map, baker map and Henon map are being applied widely owing to the advantage of simplicity [**8, 16, 18, 19**]. However, these chaos-based image encryption algorithms are broken due to their small key spaces and weak encryption mechanisms [**2**, **7, 13, 15]**. A good encryption scheme should be sensitive to large space cipher keys, resist brute-force attack, should possess good statistical properties to handle statistical attack, differential attack, known plaintext attack, and chosen-plaintext attack. To incorporate these qualities researchers had improved chaos-based cryptosystems with large key spaces and good permutation-diffusion mechanisms. 3D chaotic baker maps[**4, 5**] with modifications to the conventional baker map, Piecewise nonlinear chaotic algorithm[**9**], 3D Arnold cat maps[**3**, **20**], 3D Henon chaotic map [**12**], Hyper-chaotic differential systems[**10**], cellular automata[**11**] and multi chaotic systems **[14, 17**]were devised to improve chaos-based cryptosystems. Though the strengths and weakness between the existing encryption schemes vary, the common problem is real number arithmetic computations. Table1 shows few chaos-based algorithms. There are many problems that make the applying the chaos in the image encryption very weak such as: the existence of invalid and weak keys and some keys are not sensitive for initial conditions. For this reason, in the proposed algorithm the pixel values an image is shuffled twice and the pixels values are changed once to encrypt the image and increase its space. Experimental results illustrate that the proposed image encryption scheme possesses a large key space to resist brute-force attack and also possesses good statistical properties to frustrate statistical analysis attacks.

Table 1: Real numbers Arithmetic operations used in various chaotic for defining the mapping

Chaotic Method	Mapping is defined as	
1D Logistic map	mapping, which can be defined as following $x_{k+1} = \mu_1 x_k(1 - x_k)$	$x_k \in (0, 1)$ Where $0 \leq \mu \leq 4$ is branch parameter and When 3. 5699456 $\leq \mu \leq 4$, logistic mapping will be in chaos condition.
2D Logistic map	$x_{i+1} = \mu_1 x_i(1 - x_i) + \gamma_1 y_i^2$ $y_{i+1} = \mu_2 x y_i(1 - y_i) + \gamma_2(x_i^2 + x_i y_i)$	2. 75 < μ_1<3. 4, 2. 70<μ_2<3. 45, 0. 15< γ_1 <0. 21, and 0. 13< γ_2 <0. 15, the system comes into chaotic state and x, y in the interval (0, 1).
3D Logistic map	$x_{i+1} = \lambda x_i \ (1 - x_i) + \beta y_i^2 x_i + \alpha z_i^3$ $y_{i+1} = \lambda y_i \ (1 - y_i) + \beta z_i^2 y_i + \alpha z_i^3$ $z_{i+1} = \lambda z_i \ (1 - z_i) + \beta x_i^2 z_i + \alpha y_i^3$	exhibit the chaotic behavior for 3. 53<λ < 3. 81, 0<β<0. 022, 0<α<0. 015 and x and y can take the any value between [0, 1].
Henon map	$x_{n+1} = 1 + y_n - \alpha x_n^2$ $y_{n+1} = \beta x_n$	When α =1. 4 and β = 0. 3, the system is in chaotic state. $x_0 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
Baker Map	$B(x, y) = \left(2x, \frac{y}{2}\right)$ $B(x, y) = \left(2x - 1, \frac{y}{2} + 1/2\right)$	$when \ \frac{1}{2} < x < 1$ $when \ 0 < x < 1/2$
Chaotic Method	Mapping is defined as	
2-D Zaslavskii map	$X_{n+1} = \left(X_n + v(1 + \mu Y_n) + \varepsilon v \mu \cos(2\pi X_n)\right) mod \ 1$ $Y_{n+1} = e^{-\tau} + Y_n + \varepsilon \cos(2\pi X n)$ $\mu = \frac{1 - e^{-\tau}}{\tau}$	Where, are current chaotic values and, are next chaotic values and are control parameters and is exponentiation. The key set for Zaslavskii map is {X_0, Y_0, v, ε, τ}. Commonly used values for the parameters are v= 12. 6695, ε= 9. 1, τ= 3. 0.
3D Baker Map	2x, 2y, z/2 $2x, 2y - 1, \frac{z}{4} + \frac{1}{2}$ $2x - 1, 2y, \frac{z}{4} + \frac{1}{4}$ $2x - 1, 2y - 1, \frac{z}{4} + \frac{1}{4}$	$0 \leq x < \frac{1}{2}, 0 \leq y < \frac{1}{2}$ $0 \leq x < \frac{1}{2}, \frac{1}{2} \leq y < 1$ $\frac{1}{2} \leq x < 1, 0 \leq y < \frac{1}{2}$ $\frac{1}{2} \leq x < 1, \frac{1}{2} \leq y < 1$

3. PROPOSED ALGORITHM (IEAGC)

Algorithm in **[22]** is followed to construct **COLOR_PAD** matrix where 15 colour codes are arranged in a 12x6 **COLOR_PAD** matrix. Table 2 shows the **COLOR_PAD** constructed using colour codes **24 38 39 41 4353 55 56 68 69 80 81 84 91 92.** The colour codes in **COLOR_PAD** are arranged in such a way that for any colour code C_i, none of its 16 neighbours along its diagonal is coloured by C_i.

Table 2:COLOR_PAD

56	41	69	38	68	80
39	43	24	55	81	53
55	81	53	39	43	24
80	69	41	68	38	56
68	38	56	80	69	41
24	53	81	43	39	55
43	39	55	24	53	81
41	56	38	69	80	68
69	80	68	41	56	38
81	55	39	53	24	43
53	24	43	81	55	39
38	68	80	56	41	69

The color pattern in Fig. 1 is chosen to construct the above **COLOR_PAD**

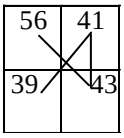

Fig. 1: Color pattern

Constructing PIXEL_VALUE matrix of color codes.
Colour codes are assigned to a **16x16PIXEL_VALUE** matrix as shown in table 3. The user at the encryption end decides the starting point(**Row_1, Col_1**)in the **PIXEL_VALUE** matrix from where the colour codes from the **COLOR_PAD** are assigned. **Row_1** and **Col_1** form a part of the secret key. The process of assigning colour codes in **PIXEL_VALUE**matrix is as follows:

Curr_Row←Row; Row_Colored←1; r←1
While (Row_Colored<= 16)
Curr_Col←Col; Col_Colored←1;c←1
While (Col_Colored<=16)
PIXEL_VALUE(Curr_Row**,** Curr_Col)
←COLOR_PAD(r, c)
Curr_Col←mod**(**Curr_Col**,** 16)+1
c←mod(c, 6)+1
Col_Colored←Col_Colored+1
end
Curr_Row←mod(Curr_Row, 16)+1
r←mod(r, 12)+1
Row_Colored←Row_Colored+1
end

Table 3:PIXEL_VALUE matrix of color codes

56	41	69	38	68	80	56	41	69	38	68	80	56	41	69	38
39	43	24	55	81	53	39	43	24	55	81	53	39	43	24	55
55	81	53	39	43	24	55	81	53	39	43	24	55	81	53	39
80	69	41	68	38	56	80	69	41	68	38	56	80	69	41	68
68	38	56	80	69	41	68	38	56	80	69	41	68	38	56	80
24	53	81	43	39	55	24	53	81	43	39	55	24	53	81	43
43	39	55	24	53	81	43	39	55	24	53	81	43	39	55	24
41	56	38	69	80	68	41	56	38	69	80	68	41	56	38	69
69	80	68	41	56	38	69	80	68	41	56	38	69	80	68	41
81	55	39	53	24	43	81	55	39	53	24	43	81	55	39	53
53	24	43	81	55	39	53	24	43	81	55	39	53	24	43	81
38	68	80	56	41	69	38	68	80	56	41	69	38	68	80	56
56	41	69	38	68	80	56	41	69	38	68	80	56	41	69	38
39	43	24	55	81	53	39	43	24	55	81	53	39	43	24	55
55	81	53	39	43	24	55	81	53	39	43	24	55	81	53	39
80	69	41	68	38	56	80	69	41	68	38	56	80	69	41	68

Colour codes in **PIXEL_VALUE** matrix are replaced by values in the range**0** and **255.** The starting point (**Row_2, Col_2**)in the **PIXEL_VALUE** matrix from where the replacement starts is supplied by the encrypting user. **Row_2** and **Col_2**form part of the secret key. The replacement process is as follows:

RGBVal←0; RowCount←1; CurrRow←Row;
CurrColorCode←**PIXEL_VALUE(Row, Col);**
CurrColorCodeIndex←1
while(ColorCodeOrder(1, CurrColorCodeIndex) <>CurrColorCode)
CurrColorCodeIndex←CurrColorCodeIndex+1
end
while(ColorCodeTried<=15)
while(RowCount<=8)
ColCount←1; CurrCol←Col
while(ColCount<=8)
if(**PIXEL_VALUE(CurrRow, CurrCol)**=CurrColorCode)
PIXEL_VALUE(CurrRow, CurrCol)←RGBVal
RGBVal←RGBVal+1
end
CurrCol←mod(CurrCol, 16)+3
ColCount←ColCount+1
end
if (CurrRow<=14)
CurrRow←mod(CurrRow, 16)+2
else
CurrRow←1
end
RowCount←RowCount+1;
end

CurrColorCode←mod(CurrColorCode, 15)+1;
ColorCodeTried←ColorCodeTried+1
While(ColorCodeOrder(1, CurrColorCodeIndex) =0)
CurrColorCode←mod(CurrColorCode, 15)+1;
ColorCodeTried←ColorCodeTried+1
end
end

Replacement of colour codes by values in the range 0 and 255 in **PIXEL_VALUE** matrix shown in table 4 starts from 7^{th} row and 5^{th} column.

Table 4: PIXEL_VALUE matrix where color codes are replaced with values in the range0 to 255

127	234	063	169	019	200	125	232	061	167	020	201	126	233	062	168
095	245	031	180	083	136	093	243	029	178	084	137	094	244	030	179
053	212	010	223	115	157	050	210	008	221	116	158	051	211	009	222
074	191	106	148	040	254	072	189	`04	146	041	255	073	190	105	147
013	161	119	194	053	224	011	159	117	192	054	225	012	160	118	193
023	130	077	237	085	170	021	128	075	235	086	171	022	129	076	236
109	215	044	151	00	202	107	213	042	149	001	203	108	214	043	150
098	248	034	183	064	138	096	246	032	181	065	139	097	247	033	182
057	197	016	228	120	162	055	195	014	226	121	163	056	196	015	227
080	174	089	133	024	238	078	172	087	131	025	239	79	173	088	132
044	154	112	206	045	216	002	152	110	204	046	217	003	153	111	205
037	142	068	251	099	184	035	140	066	249	100	185	036	141	067	250
124	231	060	166	017	198	122	229	058	164	018	199	123	230	059	165
092	242	028	177	081	134	090	240	026	175	082	135	091	241	027	176
049	209	007	220	113	155	047	207	005	218	114	156	048	208	006	219
071	188	103	145	038	252	069	186	101	143	039	253	070	187	102	144

Pixel Mapping Grid(PMG)

PMG is a table of **M** rows and **N** columns of one time pad that contains the **RGB** values of pixels in the shuffled order. Three PMG tables are constructed. **PMGR** is constructed for red panel, **PMGG** for green panel and the **PMGB** for blue panel. Procedure **PIXEL_VALUE_ASSIGN** fills these panels with the values in the range **0** and **255**. (**Row_3, Col_3**), (**Row_4, Col_4**) and (**Row_5, Col_5**) are used as the starting point by **PIXEL_VALUE_ASSIGN** in the process of filling the **PMGR**, **PMGG** and **PMGB** respectively**.** (**Row_3, Col_3**), (**Row_4, Col_4**) and (**Row_5, Col_5**) forms a part of the secret key.

Procedure **PIXEL_VALUE_ASSIGN (***PIXEL_VALUE, Row_i, Col_i, R_j, C_j, PMG***)**

//Fix **Row_i, Col_i** is fixed at **R_j, C_j** where **Row_i, Col_i** is// //thestarting point in **PMGR** or **PMGG** or **PMGB** // //fromwhere the filling process starts. Similarly **R_j, C_j**// //is the location in **PIXEL_VALUE** that holds 0**.** r2start// //andr2end is the range in **PMG**. First, **RGB** values // //are assigned to the pixels in this range. **RGB** values// //to those pixels above r2start and to those below r2end// //are assigned from the rows in the range r2start and // //r2end. r1start is the starting row in **PIXEL_VALUE**//

[Compute r2start and r2end]

// number of rowsabove **Row_i, Col_i** is more// // compared to the number of rows above **R_j, C_j** // // when **Row_i, Col_i** is fixed at **R_j, C_j** //

If (r2 >(r2-(r1-1)))

r2start← r2-(r1-1); r1start←1;

else

// number of rows above **Row_i, Col_i** is less compared // // to thenumber of rowsabove **R_j, C_j** when **Row_i, // // Col_i** is fixed at**R_j, C_j** //

r2start←1; r1start←r1-(r2-1);

end

// number of rows below **Row_i, Col_i** is less than or //// equal to M //

if((r2start+15)<=M)

r2end←r2start+15;

else

// number of rows below **Row_i, Col_i** is more than M //

r2end←M;

end

// assign RGB values to the pixels in the columns in //

// the range C2 to N of the rows in therange r2start// // and r2end //

while (c2<= N)

r1←r1start;

for r2=r2start to r2end

xxx(r2, c2)←xx(r1, c1);

r1←mod(r1, 16)+1;

end

c2←c2+1; c1←mod(c1, 16)+1;

end

//assign RGB values to thepixels in the columns in the////range C2 -1to 1 of the rows in the range r2start and // // r2end //

r1←mod(r1, 16)+1;

c1←4; c2←4; r1←1; r2←r2start;

while (c2>= 1)

r1←r1start;

for r2=r2start to r2end

xxx(r2, c2)←xx(r1, c1);

end

c2←c2-1;

if c1=1;

c1←16;

else

c1←c1-1;

end

end

[Assigning RGB values to pixels in the range r2 and M]

i←1; r2←r2+1;

while (r2<=M)

for j=1 to N

xxx(r2, j)←xxx(r2start+i-1, j);

end

i←mod(i, 17)+1;

r2←r2+1;

end

[Assigning RGB values to pixels in the range r2start-1 and 1]

i←1; r2←r2start-1;

while (r2>=1)

for j=1 to N

xxx(r2, j)←xxx(r2end-i+1, j);

end

i←mod(i, 17)+1;

r2←r2-1;

end

PMGR, **PMGG** and **PMGB** are further scrambled using the filling patterns in [22]. The sum of the direction codes participating in the filling pattern and which of the six combinations that yields the sum are added to the secret to strengthen the key space.

Finally the pixel values are changed by the following operations. In the following operations RedPanel, Green Panel and BluePanel represents RGB values of thepixels in the original image and RedPanel′, GreenPanel′ and BluePanel′ represents encrypted RGB values of the pixels in the encrypted image.

RedPanel′ (i, j) ←RedPanel(i, j)⊕PMGR(i, j);

GreenPanel′ (i, j) ←GreenPanel(i, j) ⊕ PMGG(i, j);

BluePanel′ (i, j) ←BluePanel(i, j)⊕PMGB(i, j)

The decryption algorithm is similar to the encryption algorithm but receiving encryption key and operating with theencrypted image.

4. RESULTS AND DISCUSSIONS

A general requirement for any image scrambling scheme is that the output image must be completely different from the

plain image and practically indecipherable. In this section, using MATLAB of version 7. 12. 0. 635 (R2011a) the proposed technique is applied on 560x420 Tiger image and to a 180x135 capsicum Fig. 3 different cases were analysed in detail to test the performance of the proposed technique.

A. Key-space analysis

A brute-force attack is a method of breaking an encryption system by exhaustively searching all possible keys. Its feasibility depends on the number of possible keys and on the amount of computational power available to the attacker. The key-space should be large enough to resist brute-force attacks. The proposed IEAGC algorithm makes use of twenty five integer value parameters. Each integer value requires 16 bits and as there are twenty three integer values the key-length is 400 bits and the key-space is 2^{400}. As the proposed algorithm has larger key-space it can resists brute-force attacks. Table 5 shows the key-space size of the proposed algorithm and other algorithms.

Table 5. Key-space of the proposed method and Some of the other methods in the literature

Encrypted Scheme	2-D Zaslavskii map	2D logistic	3D logistic	IEAGC
Key-Space size	2^{320}	2^{384}	2^{384}	2^{400}

B. Information Entropy analysis

Entropy is a measure of unpredictability in information content. It is used to test whether an encrypted image is random-like image with pixel values randomly distributed. In the original image, the entropy value is generally smaller than ideal value 8, because the pixel values are seldom random. Entropy reaches the maximum values when all pixel values are randomly distributed. Table 6 lists the entropy values for the original images and the encrypted images. From the results it is clear that the entropy of encrypted images is very close to the ideal value of 8. The information leakage in the proposed **IEAGC** encryption scheme is negligible and the encryption scheme is secure against the entropy based attacks.

Table 6. Entropy values for original and encrypted images

Image	Original Image	Encrypted Image
Lena	7. 5193	7. 9194
Capsicum	7. 7815	7. 9659
Baby	7. 4107	7. 8107
Tiger	7. 5769	7. 9759

C. Peak Signal to Noise Ratio analysis

PSNR which is normally used to measure images' quality losses caused by operations such as compression, noising and transmission errors is computed by comparing the original image and the processed image. In this work PSNR analysis is carried out to measure the quality of scrambled image and thereby the effectiveness of the proposed method. Thus for a good image encryption algorithm, the PSNR of the scrambled-image should be small enough for concealing the content. The PSNR value is calculated for different images and is shown in Table 7. As the PSNR value is lower it indicates that it is very difficult for attackers to get the original image from the encrypted image without the exact key.

Table 7. PSNR values for different images

Image	PSNR
Lena	14. 4453
Capsicum	11. 7890
Baby	13. 6937
Tiger	11. 1524

D. Key sensitivity

Above the 13 keys of the secret key used in [22] Row_1, Col_1, Row_2, Col_2, Row_3, Col_3, Row_4, Col_4, Row_5, Col_5, DirectionsSum and DirectionsCombination are added as the part of the secret key. An efficient encryption algorithm should be sensitive to secret key, i. e. a small change in secret key during decryption process should result in a completely different decrypted image. In the proposed algorithm, a small change in Row_1 or Col_1 or Row_2 or Col_2 or Row_4 or Col_4 or Row_5 or Col_5, DirectionsSum or DirectionsCombination results in a completely unrecognizable decrypted image. The tiger image and Capsicum image are encrypted using key 05 05 14 13 117 211 121 87 55 7214 02 and the resultant cipher-image is shown in Fig. 2b & 3c. The encrypted images are decrypted using the keys (15 0514 13117 211121 87 55 721402), (05 0504 13117 211121 87 55 721402), (05 0514 13116 211121 87 55 721402), (05 0514 13117 211101 87 55 721402), (05 0514 13117 211121 87 55 721402), (05 0514 13117 211121 87 56 721402) (05 0514 13117 211121 87 56 721412). The highlighted text refers to the change made in the key. The resultant decrypted images are shown in fig. 2 and fig. 3 are unrecognizable. Hence, it can be said that the proposed algorithm has high sensitivity to secret key. The values underlined are incorrect keys.

It is observed that all decryptions are wrong. Hence, without the knowledge of all keys, one cannot obtain the original image from its encrypted version.

a. original tiger image

b. encrypted tiger image

c. Instead of starting in 5th row of PIXEL_VALUE matrix, the decryption process started from 15th row of PIXEL_VALUE matrix started from row4thof PIXEL_VALUE matrix

d. Instead of starting from 14th row of PIXEL_ VALUEmatrix, the decryption process

e. Instead of starting from117th row of PMGR, the decryption process started from 116th row of PMGR

f. Instead of starting from 121st row of PMGG, the decryption process started from row 101 in PMGG

g. Instead of starting from55th row of PMGR, the decryption process started from 56th row of PMGR

f. Instead using the DirectionsCombination 2, the decryption process used 12

Fig 2: Key sensitivity test result for tiger image

a. original capsicum image

b. encrypted capsicum image

c. Instead of starting in 5th row of PIXEL_VALUE matrix the decryption process started from 15th row of PIXEL_ VALUE matrix

d. Instead of starting from 14th row of PIXEL_ VALUE the decryption process started from 4th row of PIXEL_VALUE Matrix

e. Instead of starting from 117th PMGR, the decryption process started from 116th row of PMGR

f. stead of starting from 121st row of PMGG, the decryption process started from d row 101 in PMGG

g. Instead of starting from55th row of PMGR, t he decryption process started from 56th row of PMGR

f. Instead of using the DirectionsCombination 2, the decryption process uses 12

Fig. 3: Key sensitivity test result for capsicum image age

E. Histogram Analysis

The original Tiger image and capsicum image and their histograms are shown in Fig. 4 and Fig. 6. The encrypted Tiger image and capsicum image and their histograms are shown in Fig. 5 and Fig. 7. It is clear from the two images shown in Fig. 5 and 7 that the proposed shuffling scheme provides more distortion and more uncorrelated adjacent pixels in the resultant shuffled image. Moreover, as we can see in Fig. 5 and Fig. 7, that the histogram of the encrypted image is fairly uniform and much different from the histogram of the plain-image shown in Fig. 4 and Fig. 6 i. e. the distribution of gray values of the encrypted image has good balance property. Hence, the encrypted image does not provide any information regarding the distribution of gray values to the attacker. As a result, the proposed algorithm can resist any type of histogram based attacks

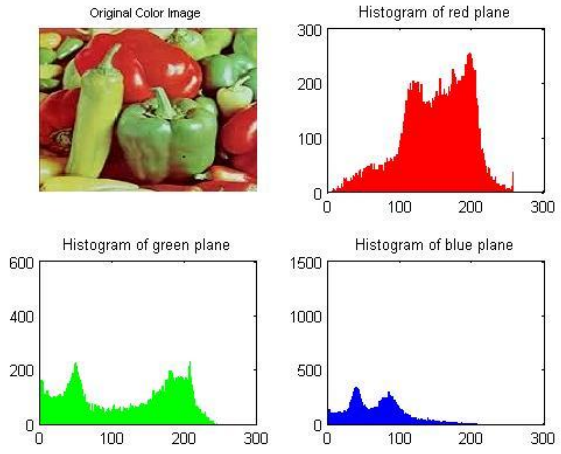

Fig. 4: Original capsicum image and its histogram

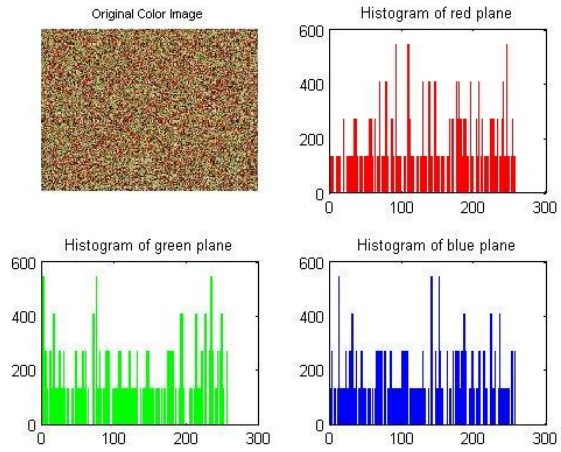

Fig. 5: Encrypted capsicum image and its histogram

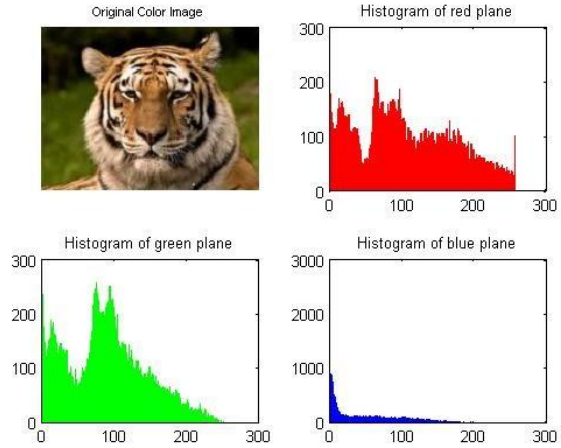

Fig. 6: Original Tiger image and its histogram

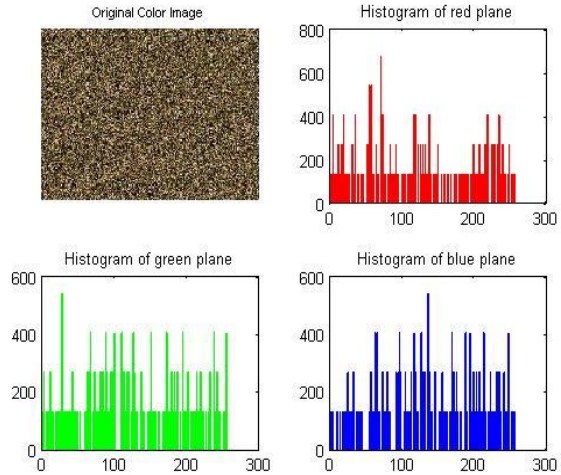

Fig. 7: Encrypted Tiger image and its histogram

F. Correlation Analysis

Correlation is a statistical analysis technique that can show whether and how strongly pairs of variables are related. The range of the correlation coefficient value is- 1 to + 1. The vertical, horizontal, and diagonal coefficients of original tiger and capsicum images and the encrypted images of tiger and capsicum are shown in Fig. 8 and 9 and the values are listed in Table 8. The values of correlation coefficients show that the two adjacent pixels in the plain-images are highly correlated to each, whereas the values obtained for cipher-images are close to 0. This shows that the proposed algorithm highly de-correlates the adjacent pixels in cipher-images.

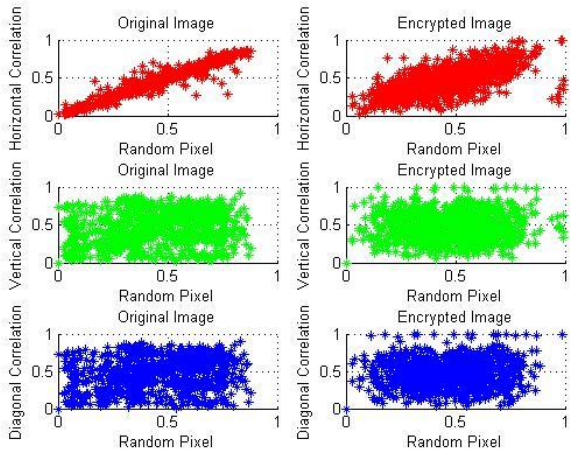

Fig. 8: Correlation of Capsicum Image

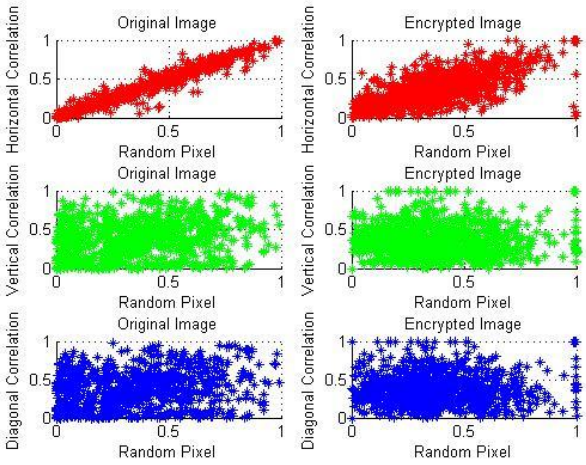

Fig. 9: Correlation of Tiger image

Table 8: Correlation coefficient

Correlation	Tiger Image		Capsicum Image	
	Original	Encrypted	Original	Encrypted
Horizontal	0. 9797	-0. 0923	0. 9883	-0. 0926
Vertical	0. 6268	-0. 0077	0. 6257	-0. 0079
Diagonal	0. 6383	0. 0654	0. 6415	0. 0664

G. Fixed Point analysis

We call it a fixed point of the scrambling scheme if its pixel coordinate does not change after scrambling. The fewer there

the fixed points of the scheme are, the more effective the scheme is, and the higher security the scheme provides. The statistical results of proportions the fixed points account for in whole pints of proposed row scrambling algorithm with randomly choosing 30000 initial values are given in table 9.

Table 9: Fixed point analysis

Image	Maximum	Minimum	Average
Lena	0. 4738	0. 4304	0. 4511
Capsicum	0. 4675	0. 4344	0. 4214
Baby	0. 4833	0. 4586	0. 4336
Tiger	0. 4588	0. 4346	0. 3900

H. Plain text Image sensitivity analysis

In plaintext analysis is a process in which cryptanalyst makes minor change in the plain text and examines the changes in the cipher text in order to gain some useful information. In the worst case scenario this attack would reveal the secret key. The two common two standards to test plain image sensitivity are Number of Pixel Change Rate (NPCPR) and Unified Average Changing Intensity(UACI). NPCR measures the percentage for the different pixels between two images and the UACI tests the average intensity of differences. The average NCPR and UACI of lena, Capsicum, baby and tiger images is shown in table 10. The experiment results show that the proposed algorithm **IEAGC** has high sensitivity to plain image sensitivity

Table 10: Plain text analysis

Lena	99. 9099	33. 3996
Capsicum	99. 7390	33. 3941
Baby	99. 8840	33. 4048
Tiger	99. 8175	33. 4388

I. Resistance to cipher-image attacks

One can damage the cipher-images by performing cropping, noising, compression, etc. on the cipher-images. In such a case, the cryptosystem's robustness against such a kind of malicious attacks is very important. A secure encryption scheme should consider the robustness against cipher-image attacks. The results of tests to cipher-image attacks are shown in Fig. 10, demonstrating that the encryption scheme is also robust against Gaussian and salt & Capsicum nosing

Original image

Gaussian noise

Salt & Capsicum noise

Fig. 10: Images with Gaussian and salt & Capsicum nosing

5. Conclusion

In this paper, the proposed algorithm changes the pixel values of the shuffled image. The security of our scheme depends on the random strategies used in generating 15 color codes, 20 color patterns, the random order for picking the blocks for encryption and decryption. As color codes, color pattern, and the random ordering of blocks are part of the secret key and there are many possibilities of randomly generating, them the algorithm is sensitive. If the attacker does not know the exact key and the attacker key has a small difference with the real key it is impossible to correctly decrypt the encrypted image. The proposed algorithm has four merits: 1) the algorithm has a large enough key space to resist all kinds of brute force attacks2) The proposed algorithm avoids arithmetic computation in changing pixel values 3) The encryption algorithm is very sensitive to the secret keys 4) the secret key size is no fixed.

References

[1] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, International Journal of Bifurcation and Chaos, 8(1998), 1259–1284.

[2] S. Li, X. Zheng, Cryptanalysis of a chaotic image encryption method, in: Proc. IEEE Int. Symposium on Circuits and Systems, vol. II, 2002, pp. 708–711.

[3] Chen, G. R., Mao, Y. B., Chui, C. K. : A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons& Fractals 21, 749--761 (2004)

[4] Mao, Y. B., Chen, G., Lian, S. G. : A novel fast image encryption scheme based on the 3D chaotic Baker map. international Journal of Bifurcation and Chaos 14, 613--3624 (2004)

[5] Yaobin Mao, Guanrong Chen, ShiguoLian, A Novel Fast Image Encryption Scheme Based On 3D Chaotic Baker Maps, International Journal of Bifurcation and Chaos, 2004, 14 (10): 3613-3624.

[6] Z. -H. Guan, F. Huang, W. Guan, Chaos-based image encryption algorithm, Physics Letters A, 346 (2005), 153–157.

[7] Gonzalo Alvarez and Shujun Li, Breaking an encryption scheme based on chaotic baker map, Physics Letters A, 352(2006), 78–82.

[8] N. K. Pareek, VinodPatidar, K. K. Sud, Image encryption using chaotic logistic map, Image and Vision Computing 24 (2006) 926–934

[9] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, A. Akhavan, A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps, Physics Letters A, 366(2007), 391-396.

[10] T. Gao, Z. Chen, A new image encryption algorithm based on hyper-chaos, Physics Letters A, 372(2008), 394–400

[11] R. Ye, H. Li, A novel digital image scrambling and watermarking scheme based on cellular automata, in: Proceedings of the 2008 International Symposium on Electronic Commerce and Security, pp. 938-941.

[12] ZhenzhenLv, Lei Zhang, and JianshengGuo A Symmetric Image Encryption Scheme Basedon Composite Chaotic Dispersed DynamicsSystem, Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT '09), 2009, pp. 191-194

[13] D. Xiao, X. Liao, P. Wei, Analysis and improvement of a chaos-based image encryption algorithm, Chaos, Solitons and Fractals, 40 (2009) 2191–2199.

[14] C. K. Huang, H. H. Nien, Multi chaotic systems based pixel shuffle for image encryption, Optics Communications, 382(2009), 2123-2127.

[15] C. Li, S. Li, G. Chen, W. A. Halang, Cryptanalysis of an image encryption scheme based on a compound chaotic sequence, Image and Vision Computing, 27 (2009), 1035–1039.

[16] RuisongYe, Wei Zhou, An Image Encryption Scheme Based on 2D Tent Map and Coupled Map Lattice, International Journal of Information and Communication Technology Volume 1 No. 8, December 2011, 344 – 348

[17] RuisongYe, WeichuangGuo, A Chaos-based Image Encryption Scheme Using Multimodal Skew Tent Maps, Journal of Emerging Trends in Computing and Information SciencesVol. 4, No. 10 2013pp 800-810

[18] Pan Tian-gong and Li Da-yong A Novel Image Encryption Using Arnold Cat International Journal of Security and Its Applications Vol. 7, No. 5 (2013), pp. 377-386

[19] Osama M., Abu Zaid, Nawal A. El-Fishawy, E. M. Nigm A Proposed Encryption Scheme based on Henon Chaotic System (PESH) International Journal of Computer Applications (0975 – 8887) Volume 61– No. 5, January 2013

[20] GuoshengGu, Jie Ling A fast image encryption method by using chaotic 3D cat maps, Optik 125 (2014) pp4700–4705

[21] Fengling Han, Xinghuo Yu ; Songchen Han Improved Baker map for image encryption, 2006. 1st International Symposium on Systems and Control in Aerospace and Astronautics, pp 1276-79

[22] D. Sathya srinivas, E. KarthikeyanA Novel Image Scrambling Algorithm using Graph Coloring, International Journal of Applied Engineering Research, Volume 10, Number 15, 2015, pp 35131-35149