A New RGB Image Encryption based on a Combination of 2D Chaotic Maps

Dina Riadh Alshibani
Assistant Lecturer, Department
of Computer Science,
University of Al Mustansiriyah

ABSTRACT
In this paper a new approach for RGB image encryption is based on a blend of 2D chaotic maps. Chaotic Henon map, chaotic Burger map and chaotic Gingerbreadman map is used in order to meet up the requirements of the protected image transmission. The proposed image encryption scheme is composed of two Transformation processes and two substitution processes. Different statistical methods, such as correlation coefficient, entropy, and histogram provide analysis of the efficiency of the proposed image encryption method.

Keywords
Henon map, Burgers map, Gingerbreadman map, permutation, subsitution, Chaotic map.

1. Introduction
The integrity and security of digital image information against illegal copying and distribution has become crucial issue with the fast developments in digital image processing and network communication.

Image ciphering, is a helpful means to protect images and videos to assurance the integrity of private or copyrighted materials. But, it is a tough task because it is quite diverse from manuscript encryption due to the inherent properties of images such as mass data volume, well-built pixel correlation. Which are usually hard to handle by using usual techniques. Nevertheless, various new image encryption schemes have been suggested in recent years, among which the chaos-based approach appears to be a promising direction.

The positions manipulation of the plain image pixels followed by altering the gray values of the shuffled pixels is used In order to improve the security performance of the proposed image encryption algorithm, in this paper, three of 2D chaotic maps are used to implement the security of the proposed encryption process [1, 2].

2. CHAOTIC MAPS:
Currently, chaotic maps are excessively used for protected or secretive communications because of their fantastic features, such as sensitivity towards primary values and parameter, non-periodicity, random-like behavior, non-convergent and mixing property etc., Because of these features, chaos theory has been a good selection for image encryption which is well-matched with the critical necessities in cryptography like confusion and diffusion.

Chaotic map is used to construct the chaotic series and used to systematize the encryption procedure. The chaos streams are generated by using a diversity of chaotic maps. Among the various maps, three 2D maps are investigated and their uniqueness is analyzed [2, 3].

2.1 Henon Map
Is a two dimensional discrete time chaotic system. It is defined as:

$$xhen(k) = 1 - ahen * xhen(k-1)^2 + bhen * yhen(k-1)..(1)$$

$$yhen(k) = xhen(k-1)..(2)$$

Where $ahen$ and $bhen$ are non-zero parameters. This map evolve chaotically when $ahen$ =1.4, $bhen$ =0.3. Details of Henon map can be found in [4].

2.2 Burgers Map
Is a two-dimensional discrete time hyper-chaotic system. It is defined as:

$$\mathrm{xber(i)} = \mathrm{aber} * \mathrm{xber(i-1)} - \mathrm{yber(i-1)}^2 \quad .. \quad (3)$$
$$\mathrm{yber(i)} = \mathrm{bber} * \mathrm{yber(i-1)} + \mathrm{xber(i-1)} * \mathrm{yber(i-1)}.. \quad (4)$$

Where aber and bber are non-zero parameters. This map evolve chaotically when aber =0.9, bber =0.856. Details of Burgers map can be found in [5].

2.3 Gingerbreadman map
In dynamical systems theory, the Gingerbreadman map is a chaotic 2D map. It is defined as:

$$xgin(i) = 1 + abs\big(xgin(i-1)\big) - ygin(i-1)..(5)$$
$$ygin(i) = xgin(i-1)..(6)$$

Details of Gingerbreadman map can be found in [6].

3. THE PROPOSED SYSTEM:
The Proposed Image Encryption Algorithm is for the most part comprises of two stages. So as to dispose of the relationship of image pixels in closest region the pixel positions is blended this present the first stage. The second stage will be performed by change of pixel dark qualities. Figure 1 demonstrates the general structure of the proposed picture encryption and unscrambling calculation. The base line of the proposed image encryption system can be summarized as follows:

1. Convert image into 1D format using zigzag scan.
2. Separating 1D of input image into blocks (same size blocks).
3. Mixing stage:
   - First level:
     - Generation of Transformation key1.
     - Transformation the blocks with the Transformation key1.
   - Second level:
     - Generation of Transformation key2.
     - Transformation the pixels within each block with the Transformation key2.
4. Confusion stage:

- Third level
  - Generation of mask key1.
  - Application of addition operation between mask key1 and the mixed image pixels.
- Fourth level:
  - Generation of mask key2.
  - Application of XOR operation between mask key2 and image pixels resulted from third level.

5. Blend the blocks
6. Convert to 3D format

The general structure of the proposed image decryption design can be summarized as follows:

1. Convert image into 1D format.
2. Separating of input image into blocks.
3. Inverse confusion level:
   - First level
     - Generation of mask key2.
     - Application of XOR operation between mask key2 and encrypted image pixels.
   - Second level:
     - Generation of mask key1.
     - Application of inverse addition operation between mask key1 and the image pixels result from first level.
4. Inverse mixing level:
   - Third level:
     - Generation of Transformation key2.
     - Inverse Transformation the pixels within each block with the Transformation key2.
   - Fourth level:
     - Generation of Transformation key1.
     - Inverse Transformation the blocks with the Transformation key1.
5. Blend the blocks
6. Convert to 3D format.

3.1 Generation of Transformation Keys:

To generate transformation Key1, firstly initiate $N \times N \times 3$ chaotic values of 2D Henon Map $(xhen, yhen)$, and two of 2D Burger map $(xber1, yber1), (xber2, yber2)$. where the X (xber1) crop of the first Burger map is supplied to the X of the second Burger Map as introductory status and the Y (yber1) crop of the first Burger map is supplied to the Y of the second Burger Map as introductory status. Then $(N \times N \times 3)/256$ values of Key1 are generated as:

$$\text{Key1}_i = \begin{cases} \text{xber1}_i, & xhen < yhen \\ \text{yber2}_i, & xhen \geq yhen \end{cases} \quad \dots (7)$$

Here, Key1_i is a concoction crop concatenation of xber1 and yber2 , and the two crops of 2-D Henon maps xhen and yhen acts as a manage key to pick whichever xber1 or yber2 of Burger map. key1 is used for block Transformation. The following matlab function is used to illustrate the steps of generating the Transformation arrays for block and pixel Transformation:

```
function perm_matrix=blk_Transformation(choatic_matrix)

[r c]=size(choatic_matrix);
s=r*c; index=[1:s]; temp=0; temp_index=0;
l=1;
   for i=1:s
     for j=i+1:s
       if  choatic_matrix(i)<choatic_matrix(j)
          temp=choatic_matrix(i);

          choatic_matrix(i)=choatic_matrix(j);
          choatic_matrix(j)=temp;
          temp_index= index(i);
          index(i)=index(j);
          index(j)=temp_index;
        end
    end
  end
 perm_matrix=index;
end
```

The following matlab function is used to explain the block Transformation process:

```
function
perm_img_blk=blk_Transformation(img,perm_matrix_blk)
 [r c]=size(img);
 s=r*c;
perm_img=zeros(size(s));
for i=1:s/256
   temp_key=perm_matrix(i);


perm_img(256*temp_key255:256*temp_key)=img(256*i255:
256*i);
 end
end
```

To generate Transformation Key2, $N \times N \times 3$ values of Key2 are generated as:

$$\text{Key2}_j = \begin{cases} \text{yber1}_j, & xhen \geq yhen \\ \text{xber2}_j, & xhen < yhen \end{cases} \quad \dots (8)$$

Here, Key2_j is a concoction crop concatenation of yber1 and xber2 , and the two crops of 2-D Henon maps xhen and yhen acts as a manage key to pick whichever yber1 or xber2 of Burger map. The steps of pixel Transformation is provided in the following matlab function:

```
function perm_img_pxl=pixel_per(img, per_pxl_matrix)
 [r c]=size(img);
s=r*c;
perm_img_pxl=zeros(size(per_pxl_matrix));
l=1;
for i=1:r:r*c
   for j=i:r*l
     temp_key=per_pxl_matrix(j);
     perm_img_pxl(j)=img(temp_key);
   end
    l=l+1;
end
end
```

Figure 2 show the Generation of the two Transformation Keys.

3.2 Generation of Masking Keys:

Figure 3 explain the Generation of the Mask Keys. As a start, generate $N \times N \times 3 \times 8$ chaotic values of xhen, yhen, xber, yber, xgin, ygin, the generated chaotic concatenations are a factual valued concatenations. These factual valued concatenations can be transformed into binary concatenations as following:

$$\text{mat1}_k = \begin{cases} 1, & xhen \geq yhen \\ 0, & xhen < yhen \end{cases} \quad \dots (9)$$

$$\text{mat2}_k = \begin{cases} 1, & xber < yber \\ 0, & xber \geq yber \end{cases} \quad \dots (10)$$

$$\text{mat3}_k = \begin{cases} 1, & xgen \geq ygen \\ 0, & xgen < ygen \end{cases} \quad \dots (11)$$

The generated random binary concatenations are used to generate two masks as following:

$$\text{key1} = \text{mat1} \otimes \text{mat2} \quad \dots (12)$$

$$\text{key2} = \text{mat1} \otimes \text{mat3} \quad \dots (13)$$

Where $\otimes$ denoted the xor operation. Then Key1 and Key2 concatenations are then abbreviated by bringing together each consecutive 8 bits into one decimal value.

Figure 4 is used to illustrate the preprocessing performed on each consecutive 8 bits before convert them into integer value.

Masking operation is performed by following formula:

$$\text{enc}_{\text{img (i)}} = \left(\left(\text{shu}_{\text{img (i)}} + \text{Key1(i)} \right) \bmod 256 \right) \oplus \text{Key2(i)} \oplus \text{enc}_{\text{img (i-1)}} \quad \dots (14)$$

Where i= 1, 2. . . N×N×3, $\text{shu}_{\text{img (i)}}$is the transformed image, Reshape 1D concatenation enc_img(i) to a color encrypted image. The decryption progresses similar to the encryption one, described above, but in the reverse order.

4. SECURITY ANALYSIS:

In this section, the performance of the proposed image encryption scheme is analyzed in detail.

The discussion was made of security analysis of the proposed image encryption scheme including some important ones like statistical sensitivity, key sensitivity analysis, key space analysis etc. to prove the proposed cryptosystem is secure against the most common attacks.

The outcomes of the performance of the proposed image encryption are showed as Fig. 5.

4.1 Histogram Analysis:

The histogram of the plain Image and its Encrypted one are displayed in Figure 6. It demonstrates that the pixel gray value of the encrypted image is extending in the whole pixel quality space. Therefore it shows that the proposed image encryption calculation has great attributes of gray equitably circulation. In that way, it can battle against certain level of fact investigation assault.

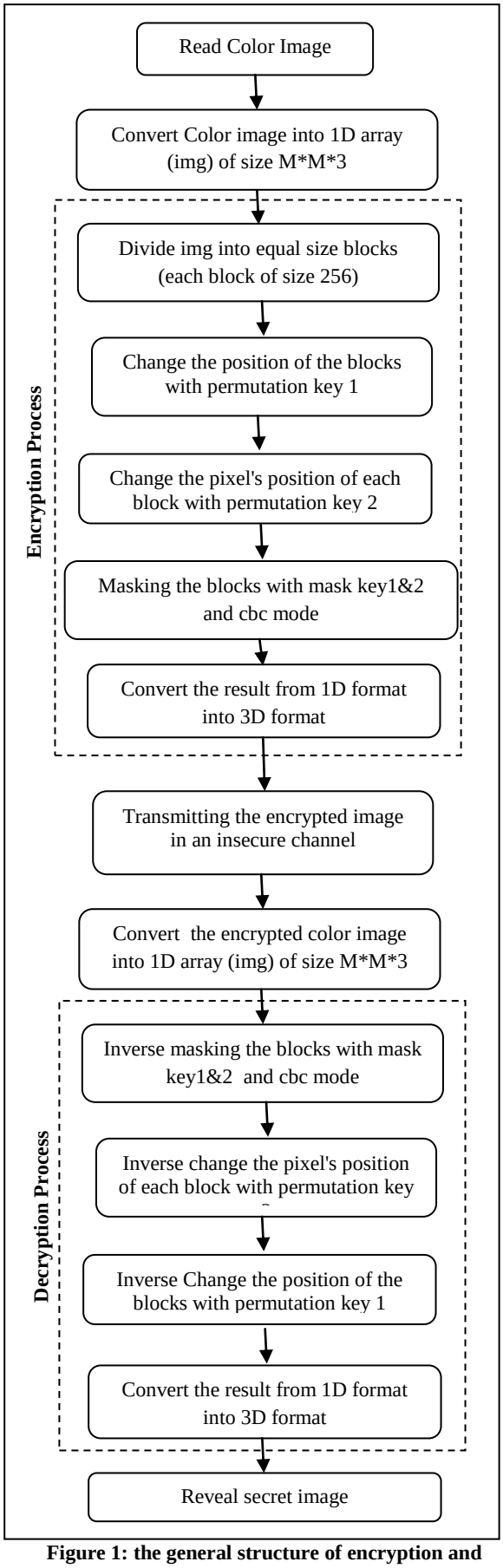


Figure 1: the general structure of encryption and decryption Processes

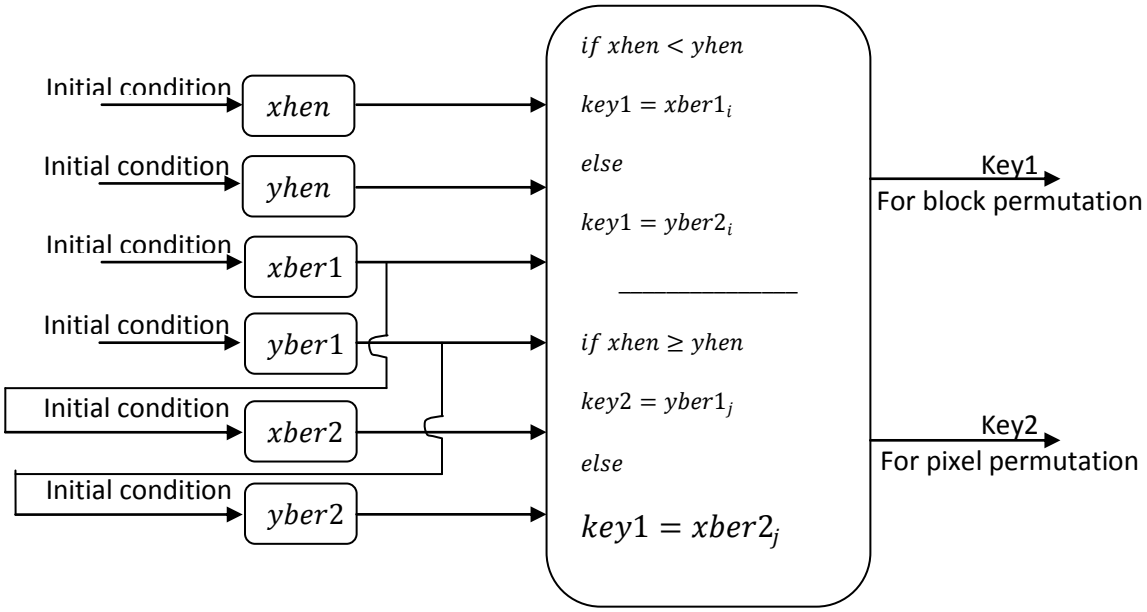


Figure 2: The Generation of the two Transformation Keys.

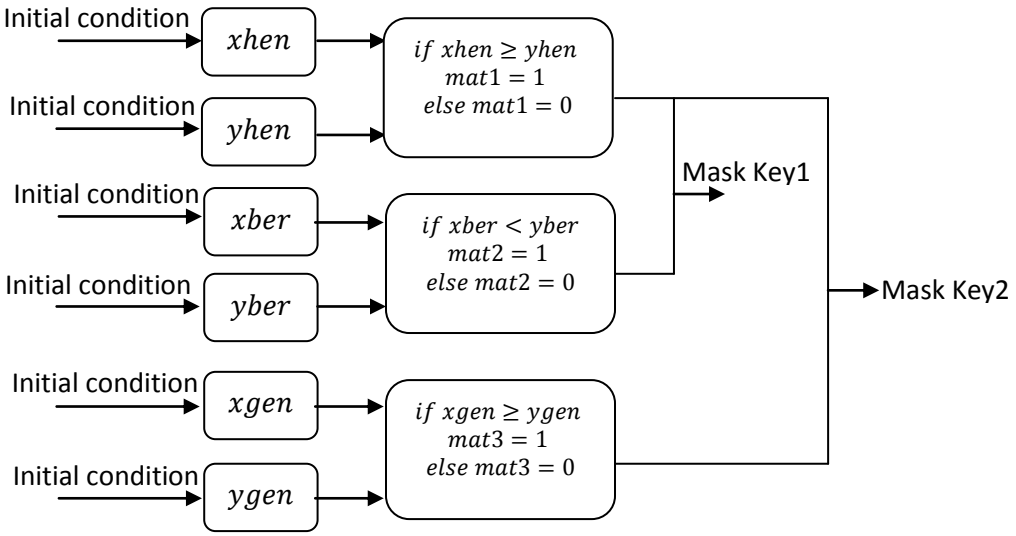


Figure 3: The Generation of the two Masking Keys.

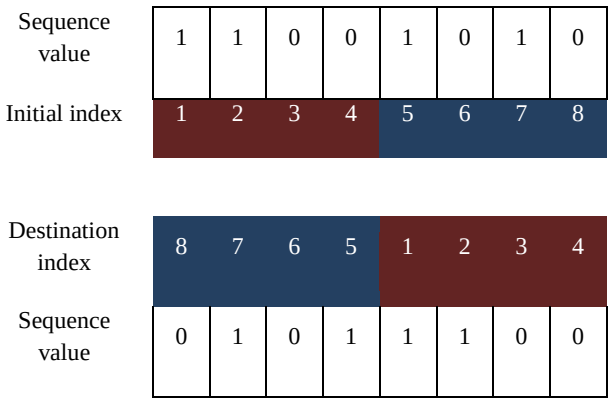
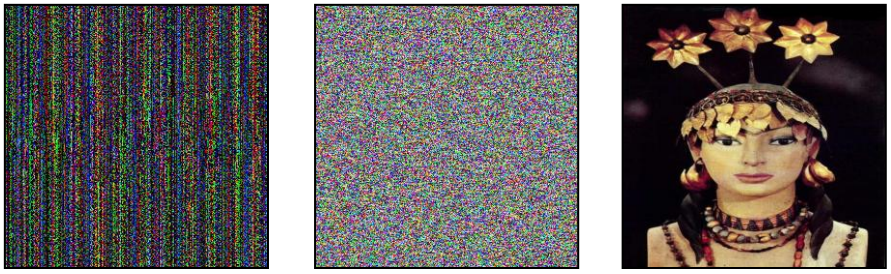


Figure 4: The per-processing performed on each consecutive 8 bits before convert them into integer value.

(A) Original Image **(B) Zig-zag Scan** **(C) Block Permutation**



(D) Pixel Permutation **(E) Encrypted Image** **(F) Reconstructed Image**

Figure 5 : Image Encryption Algorithm Result

4.2 Correlation of adjacent pixels: To check the relationship between two adjoining pixels, the accompanying exhibitions are completed. In the first place, we choose 5000 sets of two adjoining pixels arbitrarily from a image and afterward ascertain the relationship coefficient of the chose sets utilizing the accompanying formula:

$$Cr = \frac{cov_{x,y}}{\sqrt{D_x}\sqrt{D_y}} \dots (15)$$

$$cov_{x,y} = \frac{1}{T}\sum_{i=1}^{T}\big(x_i - E(x)\big)\big(y_i - E(y)\big) \dots (16)$$

$$E(x) = \frac{1}{T}\sum x_i \ , D(x) = \frac{1}{T}\sum\big(x_i - E(x)\big)^2 \dots (17)$$

Where x,y are the grey-scale values of two adjacent pixels in the image and T is the total pairs of pixels randomly selected from the image [7]. The correlations of two adjacent pixels in the plain-image and in the cipher-image are shown in the Table 1.

It is clear table 1, that the correlation coefficients of the adjacent pixels in the plain and the cipher images are far apart, that means the cryptosystem success to convert the high correlation coefficients values from the plain image into very little correlation coefficients between adjacent pixels in the cipherimage.

4.3 Entropy Information: Information entropy is defined to express the degree of uncertainties in the system. It is well known that the entropy H(m) of a message source m can be calculated as:

$$H(m) = \sum \ P(m_i)\ \log_2 \frac{1}{p(m_i)} \dots (18)$$

Where p(mi) represents the probability of symbol mi and the entropy is expressed in bits. When the messages are encrypted, their entropy should ideally be 8.

Ideal entropy of an encoded picture ought to be equivalent to 8, which relates to an arbitrary source. Basically, perfect data entropy can't be attained. It is dependably less than the perfect quality [8]. The qualities computed in Table 2 are near the perfect value.

4.4 Key Space Analysis:

In the proposed system, all introductory status and control parameters compose the secret key of encryption algorithm. For a 10^{-9} floating point precision, all keys parameters (four introductory status, six control parameters to Transformation key and six introductory status , six control parameters to mask key) can take 10^9 possible values. Therefore, the key space comes out as $((10^9)^{22}$, which are big as much as necessary to defend against the brute force-attack. The proposed image encryption Algorithm has $(10^9)^{22}$ different combinations of secret key. An image cipher with such a large key space is satisfactory for trustworthy practical use and can resist all kinds of brute force attacks.

4.5 Key affectability analysis and differential attack:

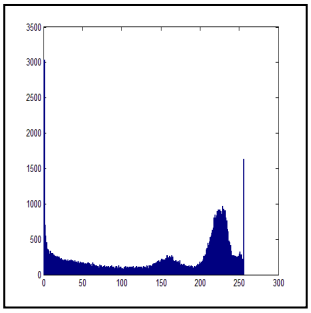
Key affectability is amazingly vital for image encryption plans. Key affectability implies the change of a solitary bit in the secret key ought to deliver totally diverse encoded image. The key affectability can be evaluated by utilizing and parameters as given beneath.NPCR (Number of Pixels Change Rate) involve the alteration average of the amount of pixels of the figure image when stand out pixel of the plain image is changed. NPCR is calculated with the following formula [9]:

$$NCPR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \dots (19)$$

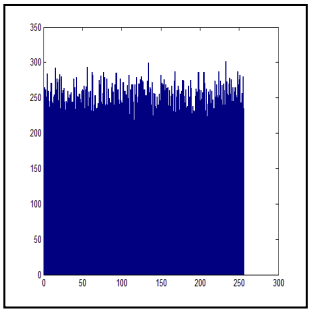
The bound together ordinary developing power (UACI) measures the ordinary force of contraindication between the plain image and figured image. UACI is calculated with the following formula [10]:

$$\mathrm{UACI}(\mathrm{p},\mathrm{s}) = \left(\frac{1}{n}\sum \frac{|p_i - S_i|}{2^{32}-1}\right) \times 100\% \ldots (20)$$

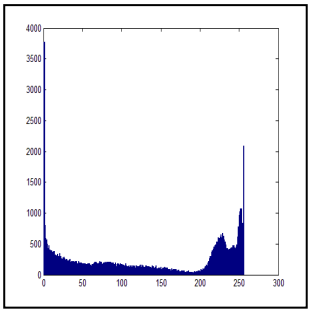
Table 3, showing The NPCR and UACI values. The results shows that a swift change in the original image will result in a significant change in the ciphered image, therefore the algorithm proposed has a good ability to differential attack.



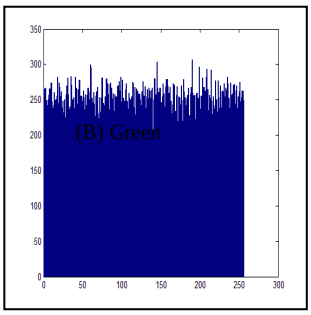
(A) Red



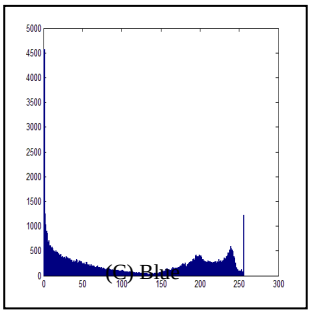
(A) Red



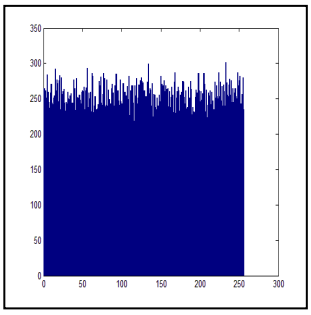
(B) Green



(B) Green



(C) Blue



(C) Blue

Figure 6: Histogram of original and encrypted image

Table 1: The correlation of adjacent pixels

case	direction	Original Image Correlation		
		Red	Green	Blue
1.	Vertical	0.7969	0.8054	0.8165
	Horizontal	0.8154	0.8186	0.8012
	Diagonal	0.7706	0.7878	0.7807
2.	Vertical	0.8392	0.8704	0.9077
	Horizontal	0.9377	0.9355	0.9513
	Diagonal	0.8857	0.8843	0.9170
3.	Vertical	0.9286	0.9953	0.9669
	Horizontal	0.9476	0.9656	0.9776
	Diagonal	0.9281	0.9507	0.9712
4.	Vertical	0.9621	0.9568	0.9685
	Horizontal	0.9768	0.9756	0.9827
	Diagonal	0.9578	0.9593	0.9677
5.	Vertical	0.9342	0.9586	0.9733
	Horizontal	0.9714	0.9816	0.9886
	Diagonal	0.9414	0.9961	0.9748
6.	Vertical	0.9482	0.9323	0.9170
	Horizontal	0.9734	0.9671	0.9491
	Diagonal	0.9425	0.9351	0.9096
case	**direction**	**Encrypted Image Correlation**		
		Red	**Green**	**Blue**
1.	Vertical	0.0044	-0.0097	-0.0191
	Horizontal	0.0112	0.0113	0.0108
	Diagonal	-0.0026	0.0028	0.0161
2.	Vertical	-0.0029	0.0014	-0.0076
	Horizontal	0.0128	0.0222	0.0165
	Diagonal	0.0210	-0.0163	-0.0277
3.	Vertical	-0.0142	-0.0201	-0.0203
	Horizontal	0.0226	0.0217	0.0252
	Diagonal	0.0024	0.0167	-0.0016
4.	Vertical	-0.0070	0.0150	-0.0156
	Horizontal	-0.0150	-0.0201	-0.0187
	Diagonal	-0.0086	0.0012	0.0176
5.	Vertical	-0.0182	-0.0219	0.0113
	Horizontal	-0.0180	-0.0216	-0.0081
	Diagonal	-0.0053	0.0270	-0.0159
6.	Vertical	-0.0023	0.0058	-0.0120
	Horizontal	0.0766	0.0740	0.0787
	Diagonal	-0.0262	-0.0163	-0.0036

Table 2: Entropy value for Original and Encrypted Image

case	Entropy Original Image			Entropy Encrypted Image		
	Red	Green	Blue	Red	Green	Blue
1.	7.3758	7.2901	7.2080	7.9970	7.9970	7.9970
2.	7.1437	7.1378	7.1323	7.9975	7.9974	7.9972
3.	6.5173	6.5919	7.0605	7.9971	7.9975	7.9977
4.	7.6690	7.6213	7.2256	7.9974	7.9972	7.9974
5.	7.6208	7.5981	7.5083	7.9975	7.9969	7.9967
6.	7.7413	7.6266	7.5387	7.9970	7.9973	7.9969

Table 3: NPCR and UACI values

case	Red		Green		Blue	
1.	**NPCR**	**UACI**	**NPCR**	**UACI**	**NPCR**	**UACI**
2.	99.59	37.59	99.64	37.15	99.58	36.76
3.	99.57	39.61	99.54	39.90	99.47	37.49
4.	99.59	30.09	99.58	29.31	99.59	29.49
5.	99.57	29.99	99.60	31.02	99.57	31.56
6.	99.66	30.03	99.58	29.78	99.56	30.13

5. CONCLUSIONS

In this paper, a new algorithm of encryption and decryption of images is planned and implemented. This algorithm is based on a combination of 2D chaotic maps. The transformation process is not satisfactory to get rid of image pixels correlations. For that reason, a masking step is extremely needed in order to weak the strong correlation among image pixels and that increases the security of the proposed algorithm. The proposed algorithms is very responsive to the introducer condition and control parameters that means the encrypted image cannot be decrypted correctly, if there is insignificant alter between encryption and decryption keys. Key sensitivity indicates a high security and suitability of the proposed algorithms.

6. REFERENCES

[1] A.Anto Steffi, Dipesh Sharma, " Modified Algorithm of Encryption and Decryption of Images using Chaotic Mapping", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064.

[2] Ruisong Ye, Shaojun Zeng, Junming Ma, Chuting Lai, "A Secure and Efficient Image Encryption Scheme Based on Tent Map and Permutation-substitution Architecture", I.J. Modern Education and Computer Science, 2014, 3, 19-30 , Published Online March 2014 in MECS (http://www.mecs-press.org/), DOI: 10.5815/ijmecs.2014.03.03.

[3] Nidhi Sethi, Sandip Vijay, "A Hybrid Cryptosystem for Image using Chaotic Mapping" , International Journal of Computer Science and Business Informatics, ISSN: 1694-2108 | Vol. 5, No. 1. SEPTEMBET 2013.

[4] Kai T Hansenyxand , Predrag Cvitanovi´ c, " Bifurcation structures in maps of Henon type ", 0951-7715/98/051233+29$19.50 c , 1998 IOP Publishing Ltd and LMS Publishing Ltd.

[5] Roman Senkerik, Ivan Zelinka, Michal Pluhacek and Zuzana Kominkova Oplatkova, " Evolutionary Control of Chaotic Burgers Map by means of Chaos Enhanced Differential Evolution ", INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION Volume 8, 2014.

[6] Manisha Raj , Shelly Garg, " An Innovative Approach: Image Encryption with Chaotic Maps using DNA Addition Operation ", International Association of Scientific Innovation and Research (IASIR) (An Association Unifying the Sciences, Engineering, and Applied Research), IJSWS 14-337; © 2014, IJSWS All Rights Reserved.

[7] Ruisong Ye, Weichuang Guo, "A Chaos-based Image Encryption Scheme Using Multimodal

[8] Skew Tent Maps ", Journal of Emerging Trends in Computing and Information Sciences ©2009-2013 CIS Journal. All rights reserved. Vol. 4, No.10 October 2013.

[9] ALIREZA JOLFAEI, ABDOLRASOUL MIRGHADRI, " AN IMAGE ENCRYPTION APPROACH USING CHAOS AND STREAM CIPHER ", Journal of Theoretical and Applied Information Technology © 2005 - 2010 JATIT & LLS. All rights reserved.

[10] Gururaj Hanchinamani, Linganagouda Kulakarni, "A Novel Approach for Image Encryption based on

[11] Parametric Mixing Chaotic System", International Journal of Computer Applications (0975 – 8887) ,Volume 96– No. 11, June 2014

[12] Boriga Radu, Dăscălescu Ana Cristina, Priescu Iustin and FilişCristina, "A New Fast Chaos-Based Image Scrambling Algorithm ", 978-1-4799-2385-4/14/$31.00 ©2014 IEEE.