Image Encryption Algorithm based on Chaotic Map

F. K. Tabash
Department of Computer Engineering
Aligarh Muslim University, Aligarh UP202002

M.Q. Rafiq
Department of Computer Engineering
Aligarh Muslim University, Aligarh UP202002

M. Izharrudin
Department of Computer Engineering
Aligarh Muslim University, Aligarh UP202002

ABSTRACT

In recent years, the chaos-based cryptographic algorithms have suggested some new and efficient ways to develop secure image encryption techniques. In this paper, a new approach for image encryption based on three chaotic logistic maps and multi-pseudo random block permutation has been proposed. This approach is developed to meet the requirements of a secure image transfer. In the proposed image encryption technique, the encryption process has been divided mainly into three steps; the first step is to encrypt the whole image using logistic map, the second step is to divide the image into a random number of blocks, the third step is to generate a random permutation for these blocks. Step two and three will be repeated for a fixed time of iterations. At experimental analysis, the proposed algorithm is compared with other four algorithms. The comparison results show that the proposed algorithm works more efficient than other algorithms. Furthermore, the results of several statistical analysis and key sensitivity tests show that the proposed algorithm provides an effective and secure way for real-time image encryption and transmission.

Keywords
Chaos, Logistic map, Image encryption, Algorithm.

1. INTRODUCTION

In recent years, the transmission of digital images over the Internet and wireless networks has been developed rapidly, due to the fascinating developments in digital image processing and network communications. It is necessary to protect the communicated image information against illegal usage, especially for those requiring reliable, fast and robust secure systems to store and transmit, such as military image databases, confidential video conference, medical imaging system, online private photograph album, etc [10]. However, number theory or algebraic concepts based on traditional ciphers, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA), and the algorithm developed by Rivest, Shamir and Adleman (RSA), most of which are used for text or binary data, appear not to be ideal for image encryption. Due to digital images are usually very large-sized and bulky, encrypting such bulky data with the traditional ciphers incurs significant overhead, and it is too expensive for real-time applications, which require real-time operations, such as displaying, cutting, copying, bit-rate control or recompression. In digital image, adjacent pixels often have similar gray-scale values and strong correlations, or image blocks have similar patterns. Such an extremely high data redundancy of images makes the conventional ciphers fail to obscure all visible information [13].

Fortunately, chaos-based image encryption algorithms have shown their superior performance. Chaotic systems have many important properties, such as the sensitive dependence on initial conditions and system parameters, the density of the set of all periodic points and topological transitivity, etc. Most properties are related to some requirements such as mixing and diffusion in the sense of cryptography. Therefore, chaotic cryptosystems have more useful and practical applications [7].

There are many chaos based image encryption schemes developed in literature. In 1992, Bourbakis and Alexopoulos [8] have proposed an image encryption scheme which utilizes the SCAN language to encrypt and compress an image simultaneously. Zhang et al. proposed an image encryption method based on skew tent map and permutation-diffusion architecture [12]. This method generates a P-box with the same size of plain-image and shuffles the positions of image pixels totally; it uses different key streams depending on plain-image in the diffusion process, so the method is much secure in the sense of preventing chosen-plaintext attack**.**
Zhu et al. [14] have proposed a new permutation method at the bit-level, which can confuse and diffuse the image at the same time. Liu and Wang [23] then improved the proposed scheme in [14] to encrypt color image, where the authors permuted the image at the bit-level by mixing all the bits in red, green and blue components. The chaotic map used in the permutation phase is PWLCM instead of Arnold cat map. Gao, Zhang, Liang and Li [5] have proposed a new chaotic algorithm for image encryption called NCA which uses power function and tangent function instead of linear functions where its structural parameters are obtained by experimental analysis.

Pareek, Patidar and Sud [9] have proposed image encryption technique based on two logistic maps with an external secret key of 80-bit. At the process, there are eight different types of operations used to encrypt the pixels of an image and which one of them will be used for a particular pixel is decided by the outcome of the logistic map. Fridrich [4] have demonstrated the construction of a symmetric block encryption technique based on two-dimensional standard baker map. There are three basic steps in the method of Fridrich [4]: (a) choose a chaotic map and generalize it by introducing some parameter, (b) discretize the chaotic map to a finite square lattice of points that represent pixels, (c) extend the discretized map to three-dimensions and further compose it with a simple diffusion mechanism.

The paper is organized as follows. Section one presents the current stage of chaos based image encryption work in

literature. Section two discusses the step by step the procedure of image encryption. Section three includes the experimental analysis for the proposed algorithm. Section four illustrates the security analysis of the image encryption scheme such as statistical analysis, key and plaintext sensitivity analysis, key space analysis etc. to prove its security against the most common attacks. Finally, section five gives the conclusions of the paper.

2. THE PROPOSED IMAGE ENCRYPTION PROCEDURE

In the proposed image encryption algorithm, an efficient encryption algorithm is developed by mixing two effective techniques of encryption; which are chaos-based image encryption scheme and block transformation, in such way, an ideal encryption results has obtained. In general, the algorithm depends on three logistics maps; the first one produces a pseudo-random numbers to encrypt the entire plain-image. It provides a sort of a good confusion value. The second logistic map generates two random values which are the number of columns and the number of rows that divide the image which is encrypted by logistic map 1 into a number of blocks. The third logistic map specifies a new random order for the blocks, to apply the transformation step, according to the new-random order. Then, the iteration of the step of generating random values of rows and columns to get a different number of blocks is applied and the step of generating a new random order for the new blocks. After a fixed number of iterations, a high confused and secured image are obtained. Fig.1 shows the flow diagram for the proposed algorithm. There are some abbreviations used in writing the algorithm such as LM (#) for Logistic Map (number) and Random numbers for Pseudo-random numbers generated by chaotic maps.

2.1 The Proposed Encryption Algorithm Step by Step

(1) Set the initial values and the parameter for the three logistics maps (x_1, x_2, x_3, α).
(2) Set the number of transformation iterations (*max*), i.e. how much the process of block transformation should be done.
(3) Do 100 iterations with each of the three logistic maps (LM (1), LM (2), LM (3)). At every iteration, you get a decimal fraction number of type double with 15 significant digits. At the end of this step each logistic map generates $[x_1 - x_{100}]$ double numbers. *This step was done to be sure that Logistics maps are at the chaotic region.*

(4) Encrypt the plain-image with LM (1):
  (a) Do 1 more iteration with LM (1); getting x_{101}.
  (b) If the whole plain-image is encrypted by LM (1) go to step (e), otherwise, do 1 more iteration with LM (1); getting x_{102}.
  (c) Divide x_{102} –double number of 15 significant digits- into five integers with each integer consisting of three digits.
  (d) Add the five integers to 5 bytes of the image. Take the mod of addition by 256. Output the result to intermediate image. Go step (b).
  (e) Pass intermediate image to step (5).

(5) If *count* (is a counter variable, initialized to 1) less than *max* (the number of transformation iterations) go to step (6), otherwise, go to step (10).

(6) Randomly using LM (2), specify the number of blocks that the intermediate image generated at step (4) should be divided:
  (a) Do 1 more iteration with LM (2); getting x_{101}.
  (b) Get the first digit of x_{101}, Assign it to R.
  (c) Do 1 more iteration with LM (2); getting x_{102}.
  (d) Get the first digit of x_{102}, Assign it to C.
  (e) The number of blocks = R * C.

(7) Randomly using LM (3), generate a new order for the every block, to prepare for transformation:
  (a) Do *N* more iteration with LM (3); getting $[x_{101} - x_{101+N}]$
  (b) Get the first two digits of each of *N* iterations. Assign each integer as a new position of the indexed block. For example, assume at first iteration we get 0.12345689123456 then, block (1) will be placed in position of block (12).
  (c) If all blocks have been assigned a new order which is unique and not equal to zero then go to step(d), otherwise, go to step(b).
  (d) Go to step (8).

(8) Apply transformation of the blocks to their new orders:
  (a) Specify the width and height (w, h) of each block by dividing the width and the height of the image by the number of R and C (generated in step (5)) recursively.

  (b) Specify the coordination for each block using the formula:

  (Left upper corner) $x_i = \left[\left((i \bmod c) - 1\right) * w\right] + 1$, $y_i = \left[floor(\frac{i}{c}) * h\right] + 1$

  (Right upper corner) $x_i = (i \bmod c) * w$, $y_i = \left[floor(\frac{i}{c}) * h\right] + 1$

  (Left lower corner) $x_i = \left[\left((i \bmod c) - 1\right) * w\right] + 1$, $y_i = \left[[floor\left(\frac{i}{c}\right) + 1] * h\right] + 1$

  (Right lower corner) $x_i = (i \bmod c) * w$, $y_i = \left[floor(\frac{i}{c}) * h\right] + 1$

  (c) Transform blocks into the object image with their new coordination.

(9) Add one to count. **Go to step (6).**

(10) Pass the encrypted image through public communication channel.

The process of decryption is the same of encryption but with back substitution. Next is the process of decryption:
  (1) Generate a list includes all the random values of rows and columns need for each iteration of block transformation.

  (2) Generate a two dimensions array includes all the random values of new order needed for each iteration of block transformation, i.e. first line of the array include the order of blocks for first iteration.

(3) Start transformation of the blocks with the last values of row, column and new order list.

(4) Iterate till you arrive the first iteration of transformation at the encryption process.

(5) Decrypt the image with the same process at step (4).

3. EXPERMENTAL ANALYSIS

The experimental analysis of the proposed algorithm depends on two objects as shown below.

1. USC-SIPI image database: is a collection of digitized images available and maintained by the University of Southern California, primarily to support research in image processing, image analysis, and machine vision. The database is divided into four different categories based on the basic character of the pictures. Currently, four volumes available at USC-SIPI site are textures, aerials, miscellaneous and sequences. The miscellaneous volume is chosen to measure the correlation coefficient of USCSIPI image database (freely available at http://sipi.usc.edu/ database/). The miscellaneous volume consists of 44 images out of which 16 are colored and 28 monochrome.

2. Correlation coefficient (r): is a measure of the correlation (linear dependence) between two variables X and Y, giving a value between +1 and −1. Correlation coefficient has the following properties:

- The value of r is such that -1 < r < +1. The + and – signs are used for positive linear correlations and negative linear correlations, respectively.
- Positive correlation: If x and y have a strong positive linear correlation, r is close to +1. And if r value exactly equals +1, it indicates a perfect positive fit.
- Negative correlation: If x and y have a strong negative linear correlation, r is close to -1. And if r value exactly equals -1, it indicates a perfect negative fit.
- No correlation: If there is no linear correlation or a weak linear correlation, r is close to 0.

Fig. 2 shows the experimental results with Lena BMP image. Fig. 2(a) is the colored Lena plain-image of size 256×256. Fig. 2(b) shows the encrypted image with the encryption key:

$$x_1 = 0.243312301231237,$$
$$x_2 = 0.654312355231232,$$
$$x_3 = 0.734312101284637,$$
$$\alpha = 3.99999$$

As the results show, the encrypted image is rough-and-tumble and unknowable. Fig. 2(c) is the decrypted image by use of the decryption algorithm with the same key. It can be seen that the decrypted image is clear and correct without any distortion.

In experimental analysis, the comparison between the proposed algorithm and other four image encryption techniques has been done. In the comparison, the correlation coefficient to compare between algorithms is used. According to the properties of correlation coefficient, the comparison with the absolute value of the correlation coefficient is used. The first technique used in comparison is the image encryption based on chaotic logistic map only. Then, the percentage of where the proposed algorithm gives a smaller correlation coefficient values is calculated. Table (1) shows the comparison results with the four algorithms. The results show that the proposed algorithm is working perfectly (100%) than encryption with chaotic maps only. The second technique is to use only block permutation and do comparison with the proposed algorithm, the result shows that the proposed algorithm also works very effectively (95.45%) than the block permutation techniques. The third one is the proposed algorithm by Gao, Zhang, Liang and Li [5] for image encryption called NCA which uses power function and tangent function instead of linear functions. The comparison results show that the proposed algorithm works better than NCA algorithm with percentage (70%). The fourth technique, the comparison with the algorithm proposed by Pareek, Patidar and Sud [9] where an external secret key of 80-bit and two chaotic logistic maps are employed with eight different types of operations are used to encrypt the pixels of an image and which one of them will be used for a particular pixel is decided by the outcome of the logistic map. The comparison results show that the proposed algorithm works better than the compared algorithm with percentage (88.63%). According to this discussion, it is a clear that the proposed algorithm works more effectively to get a high quality image encryption results.

4. SECURITY ANALYSIS

A good encryption procedure should be robust against all kinds of cryptanalytic, statistical and brute-force attacks. In this section, the security analysis of the proposed image encryption scheme is done. Statistical analysis, sensitivity analysis with respect to the key and plaintext, key space analysis are performed to prove that the proposed cryptosystem is more secure against the most common attacks.

4.1 Statistical Analysis

It is well known that many ciphers have been successfully analyzed with the help of statistical analysis, and several statistical attacks have been devised on them. Therefore, an ideal cipher should be robust against any statistical attack. To prove the robustness of the proposed image encryption procedure, a statistical analysis is done by calculating the histograms, the correlations of two adjacent pixels in the encrypted images and the correlation coefficient for several images, and its corresponding encrypted images of an image database as shown below.

1. Histogram Analysis: An image-histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. One example of such histogram analysis is shown in Fig.3 (a), the original image and in Fig.3 (b), (c) and (d) respectively, the histograms of red, blue and green channels of the original image in Fig.3 (a). Fig. 3 (e), show the encrypted image of the original image in Fig.3 (a) using the secret key:

$$x_1 = 0.243312301231237,$$
$$x_2 = 0.654312355231232,$$
$$x_3 = 0.734312101284637,$$
$$\alpha = 3.99999$$
$$max = 20,$$

And in Fig.3 (f), (g) and (h) respectively, the histograms of red, blue and green channels of the encrypted image are in Fig.3.

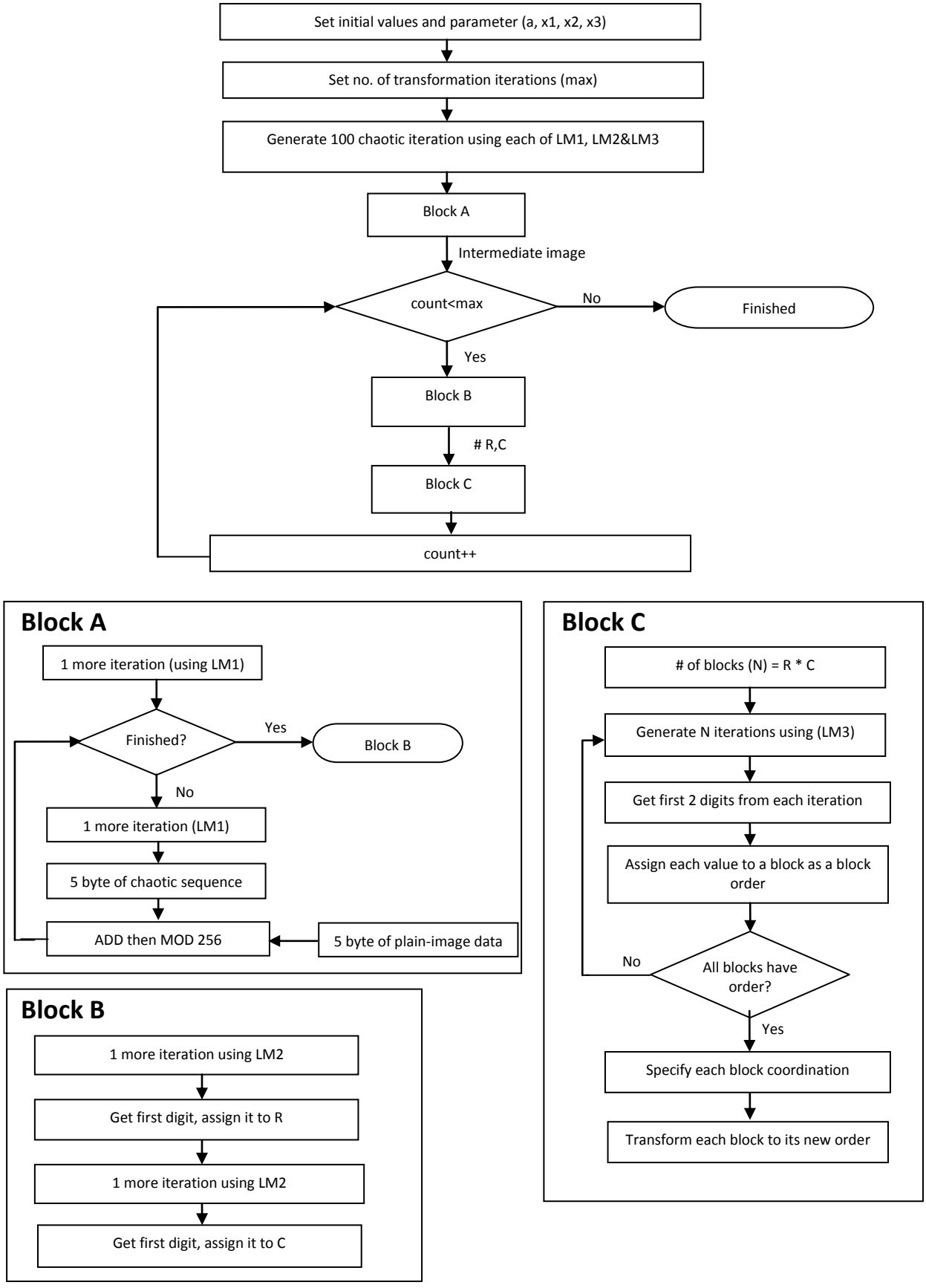

Fig 1: Block diagram of the proposed image encryption algorithm

Table 1: Comparison between the proposed algorithm and other four encryption techniques

File name	File description	Size	Type	CC-Algo1	CC-Algo2	CC-Algo3	CC-Algo4	CC-Algo5
4.1.01.tiff	Girl	256×256	Color	-0.0018	-0.0017	0.1117	0.0828	-0.0040
4.1.02.tiff	Couple	256×256	Color	-0.0010	0.0030	0.0798	0.0200	-0.0081
4.1.03.tiff	Girl	256×256	Color	0.0035	-0.0009	0.0874	0.0133	-0.0053
4.1.04.tiff	Girl	256×256	Color	-0.0012	-0.0020	0.1515	0.0404	-0.0119
4.1.05.tiff	House	256×256	Color	-0.0032	0.0043	0.1448	-0.0207	-0.0092
4.1.06.tiff	Tree	256×256	Color	0.0013	0.0040	0.1799	-0.0214	-0.0040
4.1.07.tiff	Jelly beans	256×256	Color	0.0047	0.0036	0.1227	0.1515	0.0019
4.1.08.tiff	Jelly beans	256×256	Color	0.0026	0.0020	0.1390	0.1725	0.0029
4.2.01.tiff	Splash	512×512	Color	0.0021	0.0031	0.1949	0.4030	-0.0077
4.2.02.tiff	Girl (Tiffany)	512×512	Color	-0.0053	-0.0088	0.1019	0.4846	-0.0082
4.2.03.tiff	Baboon	512×512	Color	-0.0009	0.0016	0.1562	0.0383	-0.0071
4.2.04.tiff	Girl (Lenna)	512×512	Color	0.0018	0.0020	0.1621	0.4072	0.0023
4.2.05.tiff	Airplane (F-16)	512×512	Color	-0.0007	0.0045	0.1196	-0.0109	-0.0028
4.2.06.tiff	Sailboat on lake	512×512	Color	-0.0010	0.0010	0.1828	0.0073	-0.0081
4.2.07.tiff	Peppers	512×512	Color	-0.0014	0.0047	0.1775	0.2583	0.0008
5.1.09.tiff	Girl	256×256	Gray	0.0054	0.0043	0.0866	-0.0322	-0.0101
5.1.10.tiff	Couple	256×256	Gray	0.0017	0.0057	0.1304	0.0065	0.0008
5.1.11.tiff	Girl	256×256	Gray	0.0035	0.0049	0.0965	-0.0189	-0.0077
5.1.12.tiff	Girl	256×256	Gray	0.0037	0.0053	0.1552	-0.0424	-0.0041
5.1.13.tiff	House	256×256	Gray	-0.0043	-0.0061	0.1738	0.0083	-0.0118
5.1.14.tiff	Tree	256×256	Gray	-0.0022	-0.0022	0.1215	0.0046	-0.0179
5.2.08.tiff	Jelly beans	512×512	Gray	0.0008	0.0028	0.1166	-0.0056	-0.0090
5.2.09.tiff	Jelly beans	512×512	Gray	-0.0052	0.0015	0.1086	0.0108	-0.0062
5.2.10.tiff	Splash	512×512	Gray	0.0000	0.0009	0.1510	-0.0135	-0.0060
5.3.01.tiff	Girl (Tiffany)	1024×1024	Gray	0.0003	0.0055	0.1555	0.0125	-0.0015
5.3.02.tiff	Baboon	1024×1024	Gray	0.0009	0.0003	0.0986	-0.0145	-0.0003
7.1.01.tiff	Girl (Lenna)	512×512	Gray	-0.0019	-0.0022	0.0776	-0.0230	-0.0019
7.1.02.tiff	Airplane (F-16)	512×512	Gray	-0.0019	0.0014	0.0602	-0.0177	-0.0069
7.1.03.tiff	Sailboat on lake	512×512	Gray	-0.0000	0.0003	0.0781	-0.0076	-0.0066
7.1.04.tiff	Peppers	512×512	Gray	-0.0007	0.0006	0.0998	-0.0020	-0.0112
7.1.05.tiff	Girl	512×512	Gray	0.0007	-0.0013	0.0995	0.0086	-0.0021
7.1.06.tiff	Couple	512×512	Gray	-0.0042	-0.0065	0.0923	-0.0180	-0.0066
7.1.07.tiff	Girl	512×512	Gray	-0.0001	-0.0016	0.0730	-0.0017	0.0013
7.1.08.tiff	Girl	512×512	Gray	0.0032	0.0017	0.0757	-0.0024	-0.0030
7.1.09.tiff	House	512×512	Gray	-0.0010	0.0012	0.1050	-0.0442	0.0063
7.1.10.tiff	Tree	512×512	Gray	-0.0003	-0.0008	0.0784	-0.0174	-0.0100
7.2.01.tiff	Jelly beans	1024×1024	Gray	0.0001	-0.0094	0.0643	0.0038	-0.0063
boat.512.tiff	Jelly beans	512×512	Gray	0.0009	-0.0025	0.1314	-0.0214	-0.0043
elaine.512.tiff	Splash	512×512	Gray	-0.0027	0.0006	0.1304	-0.0032	-0.0098
gray21.512.tiff	Girl (Tiffany)	512×512	Gray	-0.0026	0.0030	0.2009	-0.0115	-0.0078
house.tiff	Baboon	512×512	Color	-0.0003	0.0026	0.1529	0.0219	-0.0128
numbers.512.tiff	Girl (Lenna)	512×512	Gray	-0.0005	-0.0026	0.1689	0.0040	-0.0111
ruler.512.tiff	Airplane (F-16)	512×512	Gray	0.0030	0.0006	0.1836	0.0007	0.0031
testpat.1k.tiff	Sailboat on lake	1024×1024	Gray	0.0005	0.0009	0.1952	-0.0149	-.0007
Comparison results					**100%**	**70%**	**95.45%**	**88.63%**

Notes:
CC-Algo1: The proposed algorithm.
CC-Algo2: Image encryption based on chaotic logistic map
CC-Algo3: Image encryption based on block transformation
CC-Algo4: NCA image encryption algorithm
CC-Algo5: Pareek, Patidar and Sud proposed algorithm

(e). It is clear from Fig. 3 that the histograms of the encrypted image are fairly uniform and significantly different from the respective histograms of the original image and hence does not provide any clue to employ any statistical attack on the proposed image encryption procedure.

2. Correlation coefficient analysis: the correlation between two vertically as well as horizontally adjacent pixels in the several images and their encrypted images has been done.
Fig. 4 shows the distribution of two adjacent pixels in the original and encrypted images shown in Fig. 3a and 3e. Particularly, in Fig.4 (a) and (b), the distributions of two horizontally adjacent pixels in the original and encrypted images have been depicted respectively. Similarly, in Fig.4 (c) and (d), the distributions of two vertically adjacent pixels in the original and encrypted images have been depicted.
Table 2 shows the correlation coefficients for the original and encrypted images shown in Fig. 3a and 3e respectively. It is clear from the Fig. 4 and Table 2 that there is negligible correlation between the two adjacent pixels in the encrypted image. However, the two adjacent pixels in the original image are high correlated.

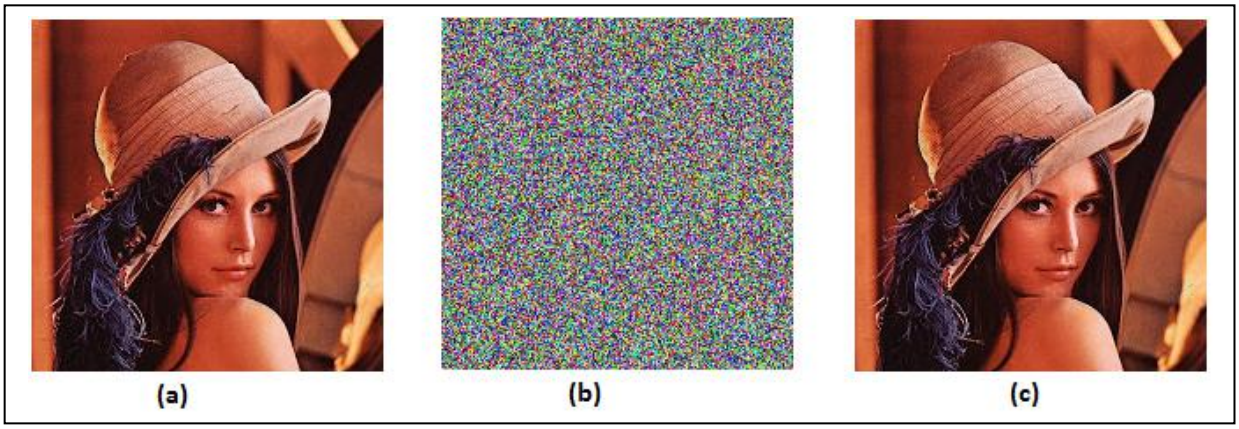

Fig. 1. Image encryption and decryption experimental result: (a) plain-image, (b) encrypted image, (c) decrypted image

Table 2: Correlation coefficients for the two adjacent pixels in the original and encrypted images are shown in Fig.3

	Original image (Fig. 3a)	Encrypted image (Fig. 3e)
Horizontal	0.9232	0.0013
Vertical	0.9679	0.0104

4.2 Sensitivity Analysis

An ideal image encryption procedure should be sensitive with respect to the secret key, i.e. the change of a single bit in the secret key should produce a completely different encrypted image. For testing the key sensitivity of the proposed image encryption procedure, the following steps are done.

(a) An original image (Fig. 5a) is encrypted by using the secret key:

$x_1 = 0.243312301231237$,
$x_2 = 0.654312355231232$,
$x_3 = 0.734312101284637$,
$\alpha = 3.99999$,
$max = 20$, and the resultant image is referred as encrypted image A (Fig. 5b).

(b) The same original image is encrypted by making the slight modification in the secret key:

$x_1 = 0.543312301231237$,
$x_2 = 0.654312355231232$,
$x_3 = 0.734312101284637$,
$\alpha = 3.99999$,
$max = 20$, (The most significant bit of x_1 is changed in the secret key) and the resultant image is referred as encrypted image B (Fig. 5c).

(c) The same original image is encrypted by making the slight modification in the secret key:

$x_1 = 0.243312301231238$,
$x_2 = 0.654312355231232$,
$x_3 = 0.734312101284637$,
$\alpha = 3.99999$,
$max = 20$, (The least significant bit of x_1 is changed in the secret key) and the resultant image is referred as encrypted image C (Fig. 5d).

(d) Finally, the three encrypted images A, B and C are compared.

Table 3 shows the results of the correlation coefficients between the corresponding pixels of the three encrypted images A, B and C. It is clear from the table that no correlation exists among three encrypted images, even though these have been produced by using slightly different secret keys.

Moreover, NPCR analysis is used to measure the sensitivity of an encryption technique. NPCR means the change rate of the number of pixels of ciphered image while one pixel of plain-image is changed. The value of the last pixel of plain-image is 22. The result was over 99%.

Table 3 Correlation coefficients between the corresponding pixels of the three different encrypted images

		Correlation coefficient
Encrypted Image A (Fig. 5b)	Encrypted Image B (Fig. 5c)	0.0048
Encrypted Image B (Fig. 5c)	Encrypted Image C (Fig. 5d)	0.0011
Encrypted Image C (Fig. 5d)	Encrypted Image A (Fig. 5b)	0.0012

4.3 Key Space Analysis

For a secure image cipher, the key space should be large enough to make the brute force attack infeasible. In the proposed algorithm, three logistics map are used and each of them has its initial conditions where a very small change on one of will give a completely different result. According to this, the key space for the algorithm will be $x_1 = [0,1]$, $x_2 = [0,1]$, $x_3 = [0,1]$, $\alpha = [3.6,4)$. The results show that the key space for the algorithm is long enough to protect the algorithm from any brute force attack.

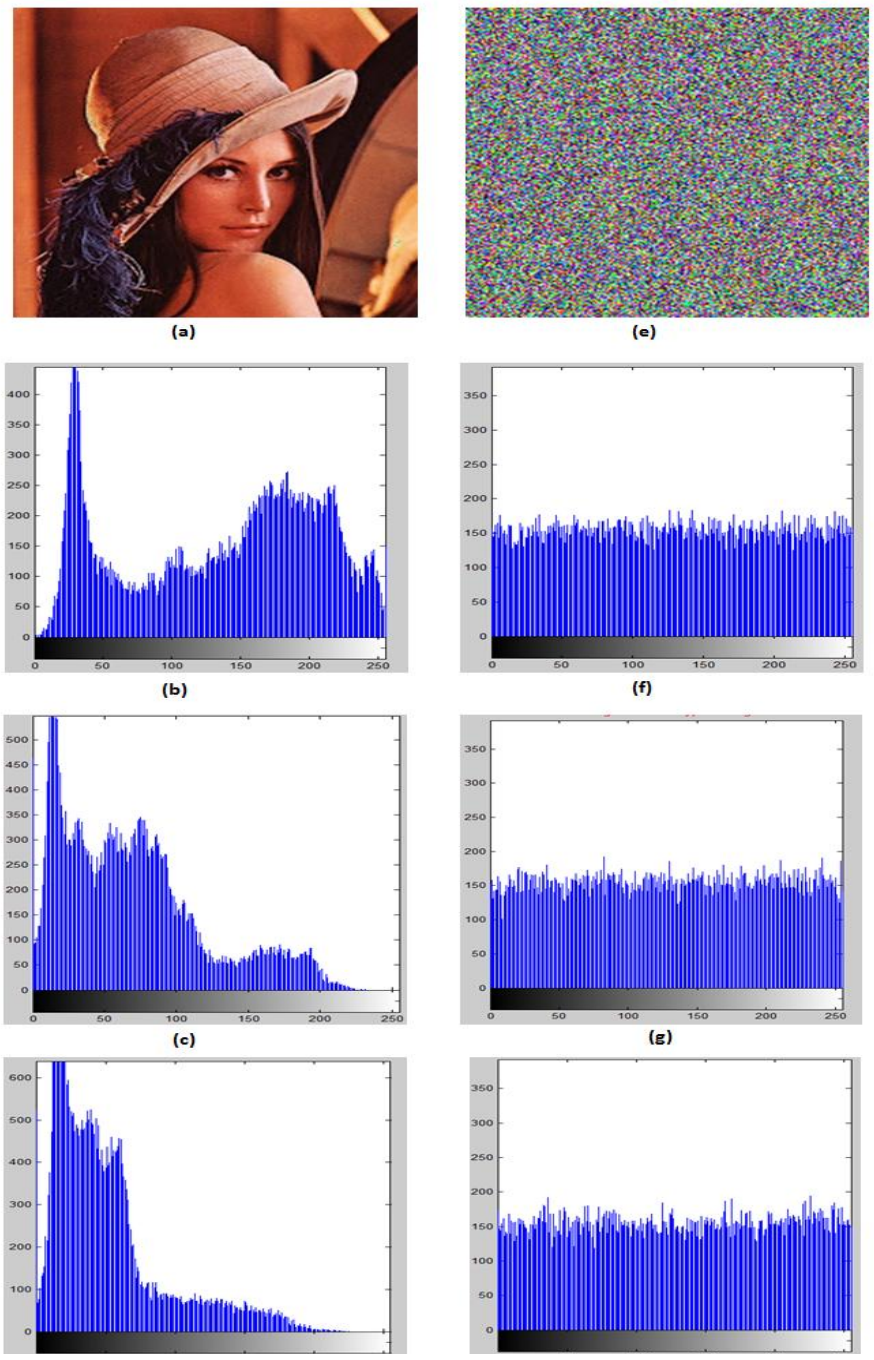

Fig. 2. Histogram analysis: (a) shows a plain image. (b), (c) and (d) respectively, show the histograms of red, green and blue channels of the plain image shown in (a). (e) Shows the encrypted image of the plain image shown in (a). (f), (g) and (h) respectively, show the histograms of red, green and blue channels of the encrypted image shown in (e).

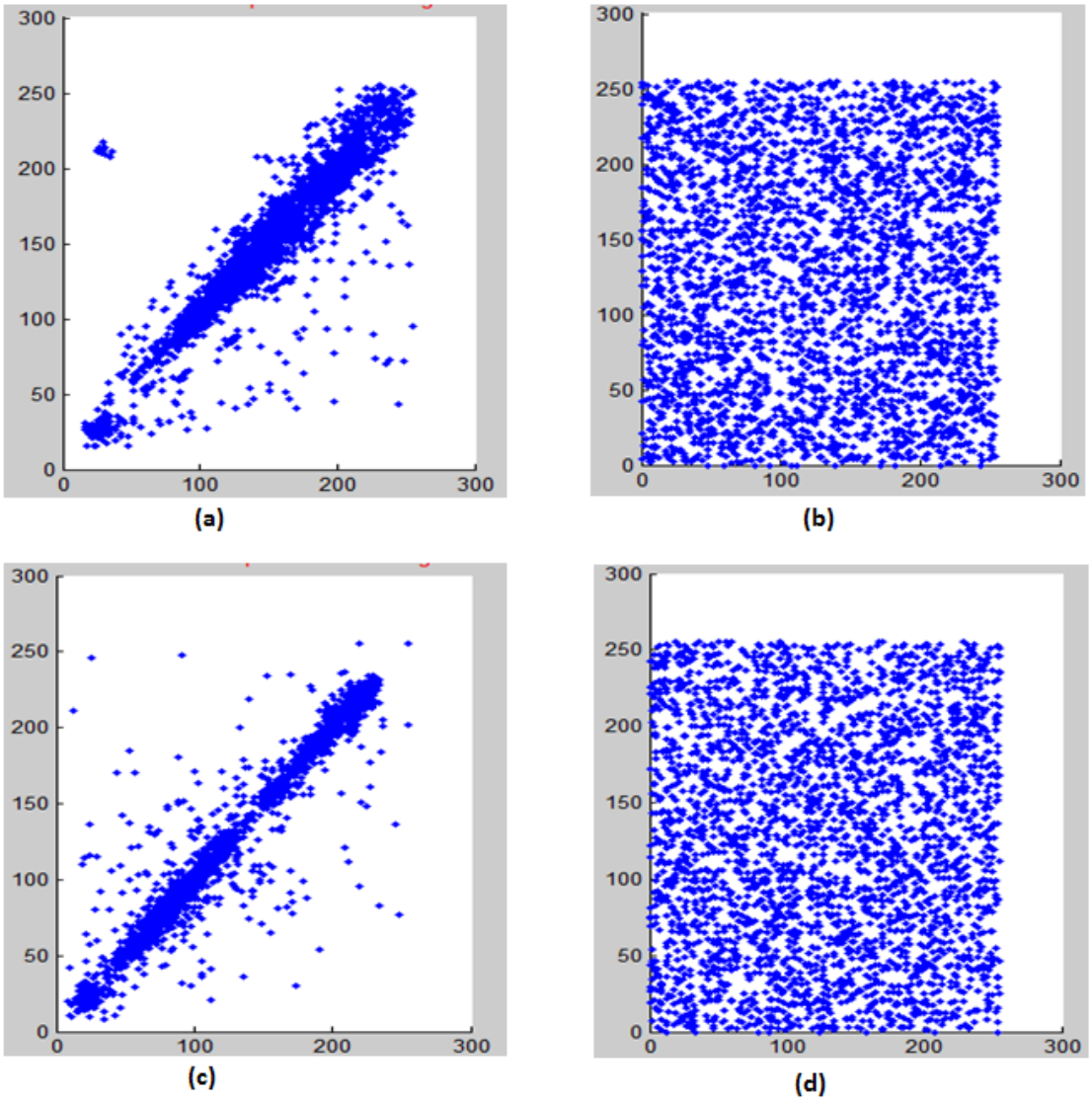

Fig. 3. Correlation of two adjacent pixels: (a) and (b) respectively, show the distribution of two horizontally adjacent pixels in the plain and encrypted images shown in Fig. 3a and 3e. (c) and (d) respectively, show the distribution of two vertically adjacent pixels in the plain and encrypted images shown in Fig. 3a and 3e.

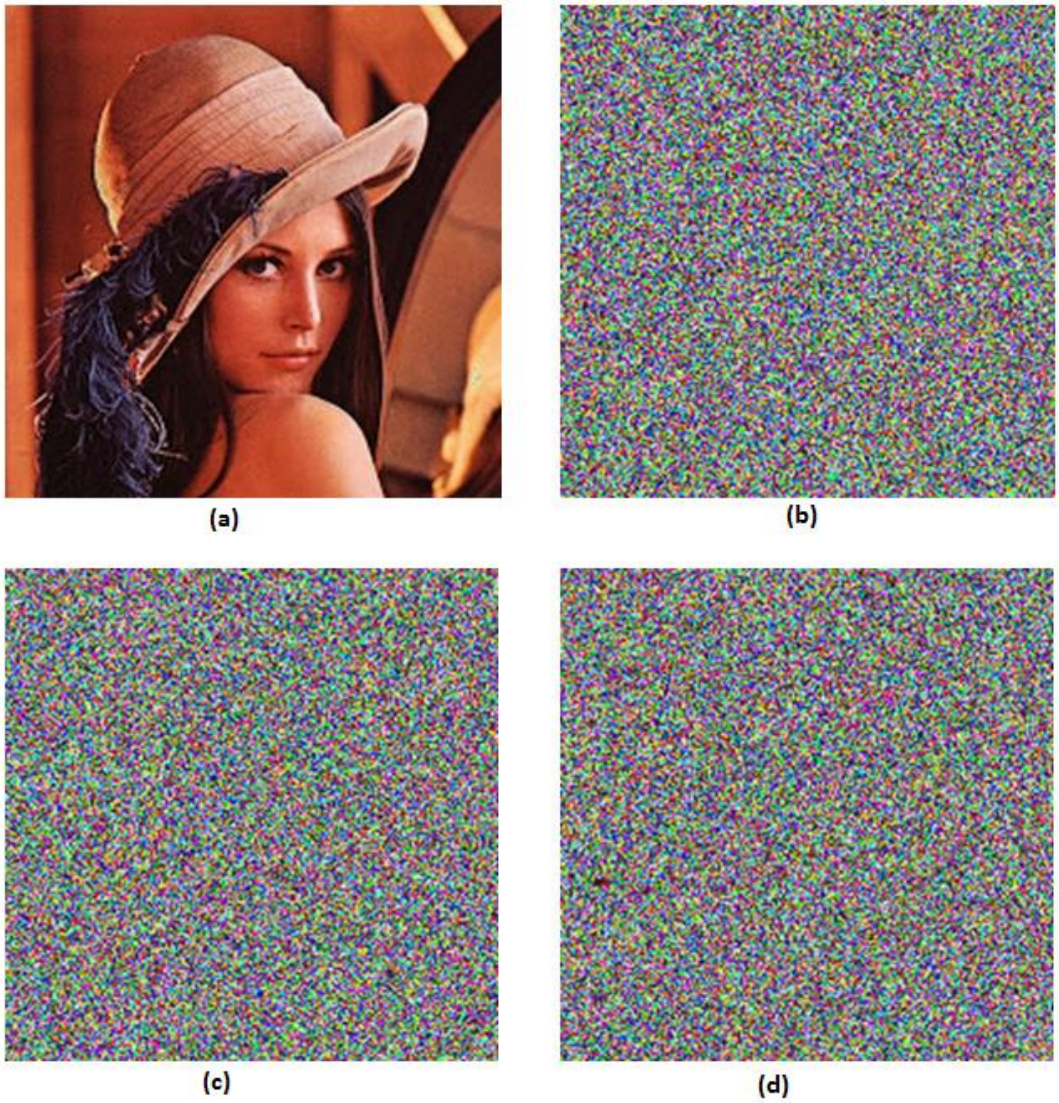

Fig. 4 Key sensitivity test: (a) shows a plain image, (b), (c) and (d) respectively, show the encrypted images of the plain image shown in (a) with slightly different keys

4.4 Time Analysis

Apart from the security consideration, running speed of the algorithm is an important factor for a good encryption algorithm. The rate of the encryption/decryption of several colored images of different sized is measured by using the proposed image encryption scheme. The time analysis has been done on Pentium-4 with 256 MB RAM computer. The average encryption/decryption time taken by the algorithm for different sized images is shown in the Table 4.

Table 4 Average ciphering speed of a few different sized colored images is used.

Image size	Average encryption/decryption time (s)
256×256	0.33
512×512	1.43
1024×1024	5.95

5. CONCLUSTION

In this paper, a proposed image encryption algorithm based on two techniques of image encryption algorithms is developed. The proposed technique depends on chaotic map encryption and iterated-random block transformation. The algorithm uses three logistic map functions to get a highly confused, diffused and secured encrypted image.

Experimental analysis shows that the proposed image encryption algorithm has some advantages of a large key space, and high-level security, as well as maintaining superior efficiency compared with similar chaotic encryption algorithms. The results of the proposed algorithm are particularly suitable for Internet image encryption and transmission applications. Furthermore, the proposed algorithm can be applied in other information security fields and unlimited only to image encryption.

6. REFERENCES

[1] Alvarez, G., and S. Li, S., (2006), some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurcation Chaos, 16, 8, pp.2129–2151.

[2] Bu SL, and Wang H. (2004), Improving the security of chaotic encryption by using a simple modulating method. Chaos, Solutions & Fractals, 19, pp.919–24.

[3] C.C. Chang, M.S. Hwang, and T.S. Chen, (2001), A new encryption algorithm for image cryptosystems, J. Syst. Software, 58,pp.83–91.

[4] Fridrich, J.,(1998), Symmetric ciphers based on two dimensional chaotic maps, Int. J. Bifurcat Chaos, 8, 6, pp1259–1284.

[5] Haojiang, G., Yisheng, Z., Shuyun, Li., and Dequn, Li., (2005). A new chaotic algorithm for image encryption. Chaos Solitons Fractals, 29, pp.393–399.

[6] Li S., and Zheng, X., (2002). Cryptanalysis of a chaotic image encryption method, IEEE Int Symp Circ Syst, 2, pp.708–11.

[7] Linhua Z., et al.,(2005), An image encryption approach based on chaotic maps. Chaos, Solitons and Fractals, 24, pp.759–765

[8] N. Bourbakis and C. Alexopoulos (1992), Picture data encryption using SCAN pattern, Pattern Recogn, 25 ,pp.567–581.

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

[10] Ruisong, Y. and Haiying ,Z.(2012). An Efficient Chaos-based Image Encryption Scheme Using Affine Modular Maps. I. J. Computer Network and Information Security, 7, pp.41-50.

[11] Wu XG, Hu HP, and Zhang BL (2004). Analyzing and improving a chaotic encryption method. Chaos, Solutions & Fractals, pp. 367–73.

[12] Zhang, et al., (2011), A novel image encryption method based on total shuffling scheme. Opt. Communication., 284, pp 2775-2780. [1-21]

[13] Zhaopin Su, Guofu Zhang and Jianguo Jiang (2012). Multimedia Security: A Survey of Chaos-Based Encryption Technology, Multimedia - A Multidisciplinary Approach to Complex Issues, Dr. Ioannis Karydis (Ed.), ISBN: 978-953-51-0216-8, InTech, Available from: http://www.intechopen.com/books/multimedia-amultidisciplinary- approach-to-complex-issues/multimedia-security-a-survey-of-chaos-based-encryptiontechnology.

[14] Zhu, Z. L., et al., (2010), A chaos-based symmetric image encryption scheme using a bit-level permutation, Information Sci, 181, pp1171-1186.