International Journal of Computer Science and Business Informatics

IJCSBI.ORG

Symmetric Image Encryption Algorithm Using 3D Rossler System

Vishnu G. Kamat
M Tech student in Information Security and Management
Department of IT, DIT University
Dehradun, India

Madhu Sharma
Assistant Professor
Department of Computer Science, DIT University
Dehradun, India

ABSTRACT
Recently a lot of research has been done in the field of image encryption using chaotic maps. In this paper, we propose a new symmetric block cipher algorithm using the 3D Rossler system. The algorithm utilizes the approach used by Mohamed Amin et al. [Commun. Nonlinear Sci. Numer. Simulat, (2010)] and Vinod Patidar et al. [Commun Nonlinear SciNumerSimulat, (2009)]. The merits of these algorithms such as the encryption structure and the diffusion scheme respectively are combined with an approach to split the key for the three dimensions to use for encryption of color (RGB) images. The experimentation results suggest an overall better performance of the algorithm.

Keywords
Image Encryption, Rossler System, Block Cipher, Security Analysis.

1. INTRODUCTION
Image encryption is relatively different from text encryption. Image is made up of pixels and they are highly correlated; so different approaches are followed for encryption of images [1-12]. One of the approaches is known as chaotic cryptography. In this approach, for encryption we use chaotic maps, which generate good pseudo-random numbers. Cryptographic properties of these maps such as, sensitive dependence on initial parameters, ergodic and random like behavior, make them ideal for use in designing secure cryptographic algorithms. Many scholars have proposed various chaos-based encryption schemes in recent years [4-12].

A scheme proposed by Mohamed Amin et al. [11] uses Tent map as the chaotic map and the scheme is implemented for gray scale images. They proposed a new approach of using the plaintext as blocks of bits rather than block of pixels. Another scheme proposed by Vinod Patidaret al.[12] uses chaotic standard and logistic maps and they introduce a way of spreading the bits using diffusion to avoid redundancy. In this paper, we propose an algorithm which utilizes the merits of the mentioned schemes. The

algorithm uses the Rossler system for the chaotic key generation. We demonstrate a way to split the 3 dimensions of the key for the 3 image channels i.e. Red, Green and Blue. The algorithm in [11] is used as a base structure and the diffusion concept from [12] is used to spread the effect of adding the key. The symmetric Feistel structure, diffusion method and key splitting of the encryption scheme provide better results.

The rest of the paper is organized as follows: Section 2 provides a brief overview of the Rossler system. Section 3 provides the algorithmic details. The results of the security analysis are shown in section 4. Lastly, Section 5 concludes the paper.

2. BRIEF OVERVIEW OF 3D ROSSLER SYSTEM

Rossler system is a system of non-linear differential equations which has chaotic properties [13]. Otto Rossler defined these equations in 1976. The equations are as given below

$$\begin{aligned} X_n+1 &= -Y_n-Z_n \\ Y_n+1 &= X_n + \alpha Y_n \\ Z_n+1 &= \beta + Z_n (X_n-\gamma) \end{aligned} \qquad (1)$$

where, α, β and γ are real parameters. Rossler system's behavior is dependent on the values of the parameters α, β and γ. For different values of these parameters the system displays considerable changes. It may be chaotic, converge toward a fixed point, follow a periodic orbit or escape towards infinity. The Rossler system displays chaotic behavior for the values of α=0.432, β=2 and γ=4.

The chaotic behavior refers to the fact that keeping the parameters constant, even a slight change in the initial value would bring a significant change in the subsequent values. For example the value of $Z_0 = 0.3$ generates the value of $Z_1 = 0.5$. After changing the value of Z_0to0.6it generates the value of $Z_1 = -1$. The same chaotic rule applies for the changes of other two dimensions (X and Y). This chaotic behavior is known as deterministic chaos, i.e. the knowledge of initial values and parameter values can help in recreating the same chaotic pattern. Hence the initial conditions have to be shared between the entities using the system for encryption/decryption process.

3. PROPOSED ALGORITHM

In this section we provide details of our algorithm. The algorithm is designed to work with color images (RGB). In this scheme the plaintext (image) is taken as blocks of bits. The block size is 8w, where 'w' is the word size which is 32 bits. Each block of data is divided and stored into 8 w-bit registers and operations are performed on them. The key length

depends on the number of rounds 'r' i.e. Key length is 4r+8. The number of rounds can vary from 1-255. We have taken 'r' to be 12 for our experimentation.

The flowchart shown in Fig. 1 displays the various steps performed on the image during the encryption process. The steps are explained in the following subsections.

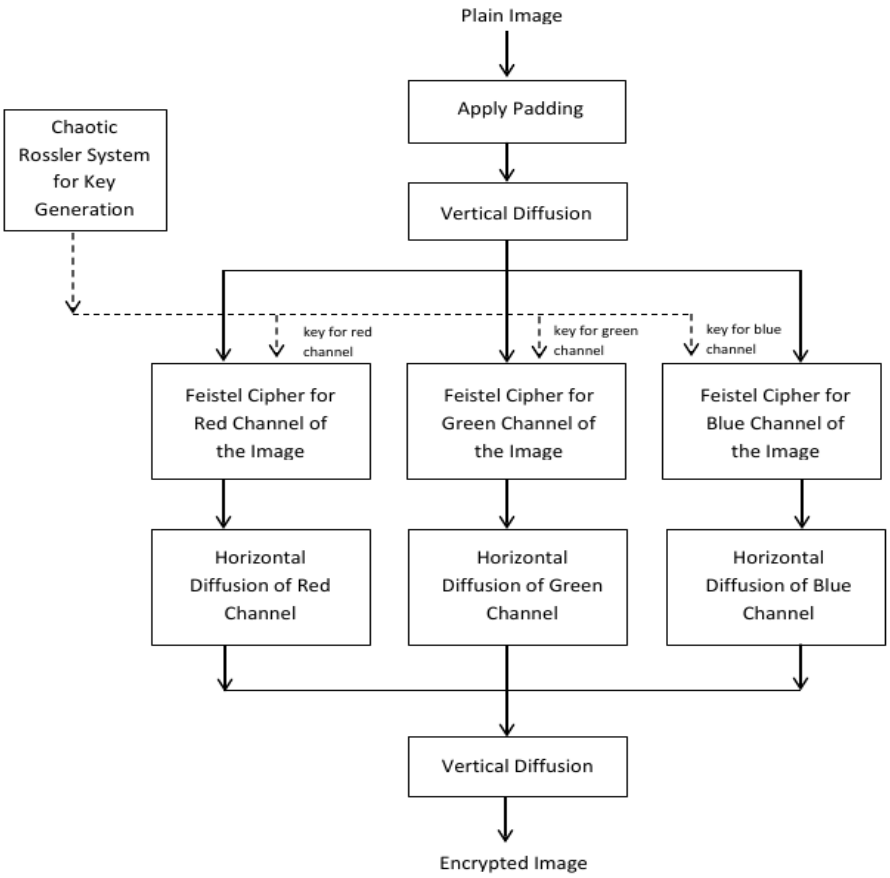

Figure 1. Flowchart of the Encryption Scheme

3.1 Padding

The processing of the image is done on block of data. 256 bits ie.32 bytes of data are encrypted/decrypted at a time using eight 32-bit registers. The image size should be a multiple of 256 bits to ensure that there is always a full block size for encryption. Hence padding is added so as to make the input block of size 32 bytes when the image size in bytes is not an integral multiple of 32. A padding of all zeros (1-31 bytes) is appended to the end of each row to make the bytes in each row a multiple of 32.

For example if the image is of dimensions 252 x 252 pixels, a 4 byte padding of zeros is appended at the end of each row. The last byte of the image then stores the number of bytes used as padding as a pixel value i.e. 4 in this case. This pixel value is used to remove the padding after decryption. After retrieving the number of bytes padded 'n', all rows are checked to determine if zeros exist in all the last 'n' bytes and in 'n-1' bytes of the last row. The padding is then removed to generate the original image.

3.2 Key Generation

The key is generated by the 3D chaotic Rossler system as shown in (1). The number of key bytes 't' depends on the number of rounds 'r' i.e. t=4r+8. We use the three equations separately. The random sequence generated by each equation of the map is used as a key separately during the encryption process of the red, green and blue channel of the image respectively. The key generation concept is as shown below. The steps repeat 't' number of times to generate necessary key bytes.

a. Iterate Rossler system of equations (1) 'r' times where 'r' is the number of rounds.

b. Use the decimal part of the X, Y, Z values to generate the key byte.

X_n = abs (X_n - integer part); // decimal part of x

Y_n = abs (Y_n - integer part); // decimal part of y

Z_n = abs (Z_n - integer part); // decimal part of z

c. Key byte for each dimension (R,G,B) is taken as X, Y, Z values respectively by mapping it to a value between 0-255.

d. For the next set of key bytes the number of iterations is changed to a value obtained by performing exclusive-or on the current set of key bytes.

Iterations for next key byte = XOR (X_n, Y_n, Z_n);

3.3 Vertical and Horizontal Diffusion

The diffusion process explained in [12] is used in the algorithm. The horizontal diffusion in our algorithm is used in a slightly different way i.e. it is performed separately on each channel after the encryption of the channel rather than using it on the entire image. The diffusion ensures spread of the key additions for the channel. The horizontal diffusion moves in the forward direction from the first pixel of a channel to the last. The second pixel is the exclusive or of first and second pixel of a channel, the third pixel is the

exclusive or of the new second pixel and the third pixel, and so on. Thus the first pixel of the channel remains unchanged.

The Vertical Diffusion is performed before and after the entire encryption and horizontal diffusion is performed on the 3 channels of the image. In Vertical Diffusion the channels are treated collectively. The processing occurs from the last pixel of the image to the first pixel. It starts by performing XOR of the green and blue values of the last pixel of the image with the red value of the second last pixel to form the new red value of the second last pixel. The green value of the second last pixel is formed by performing XOR operation on the red and blue values of the last pixel. The blue value of the second last pixel is formed by XOR operation on the red and green values of the last pixel. This continues in the backward direction. Thus the last pixel remains unchanged.

3.4 Encryption/Decryption Scheme

The encryption is performed on 256 bits (32 bytes) of data at a time using eight 32-bit registers. The algorithm is shown in Fig. 2. In the initial step four bytes of the key are added to alternate registers. 2's compliment addition is performed. Then for 'r' rounds arithmetic operations are performed on the image data. It uses a function 'f', the output of which is used as the number of rotations to be performed on another block of data. After the swapping operation of the last round, the last four key bytes are added. The entire encryption structure is displayed in Fig. 3. For decryption the algorithm follows reverse of the encryption process.

$B = B + \mathrm{K}[0];$
$D = D + \mathrm{K}[1];$
$F = F + \mathrm{K}[2];$
$H = H + \mathrm{K}[3];$
for $i = 1$ to r **do**
{
$k = ((1 - B)/4)^{*}B;$
$l = ((1 - D)/4)^{*}D;$
$m = ((1 - F)/4)^{*}F;$
$n = ((1 - H)/4)^{*}H;$
$A = (\mathrm{A} \oplus \mathrm{k}) << l + \mathrm{K}[4i];$
$C = (C \oplus \mathrm{l}) << k + \mathrm{K}[4i + 1];$
$E = (E \oplus m) << n + \mathrm{K}[4i + 2];$
$G = (G \oplus n) << m + \mathrm{K}[4i + 3];$
$(A, B, C, D, E, F, G, H) = (B, C, D, E, F, G, H, A);$ //perform swap operation
}
$A = \mathrm{A} + \mathrm{K}[4\mathrm{r}+4];$
$C = C + \mathrm{K}[4\mathrm{r}+5];$
$E = E + \mathrm{K}[4\mathrm{r}+6];$
$G = G + \mathrm{K}[4\mathrm{r}+7];$

Figure 2.Encryption Algorithm for each Channel (R,G,B)

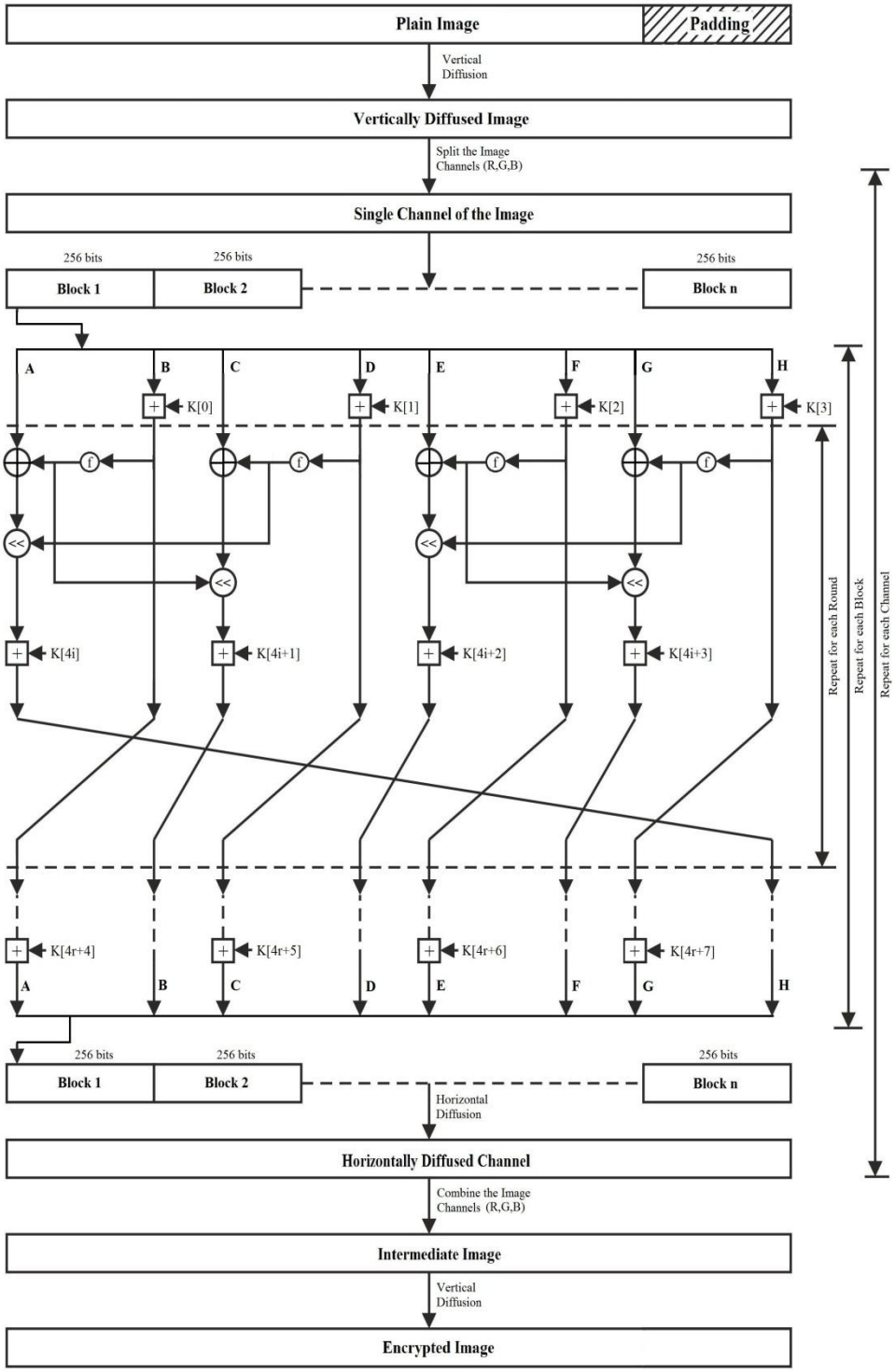

Figure 3. The Image Encryption Structure

4. EXPERIMENTATION RESULTS

We performed security analysis on six 256 x 256 color(RGB) images as shown in Fig. 4. The statistical and differential analysis tests performed display very favorable results. The results display the strength and security of the algorithm. The results have been given in [14] to demonstrate the overcoming of vulnerability in [11].

Figure 4.Plain images (clockwise from top left): Lena, Bridge, Lake, Plane, Peppersand Mandrill

4.1 Statistical Analysis

Statistical analysis is performed to determine the correlation between the plain image and the cipher image. For an encryption system to be strong the cipher image should not be correlated to the plain image and the cipher image pixels should not have correlation among them. In this section we provide the histogram and correlation analysis.

4.1.1 Histogram Analysis

When the encrypted image and the plain image do not show high degree of correlation we can consider the encryption to be secure form information leakage. Histograms are used to plot the number of pixels at each intensity level i.e. pixels having values 0-255. This helps in displaying how the pixels are distributed.

Fig. 5 depicts the histogram for the red, green and blue channels of the plain image ‘lena’ on the left side (from top to down) and the histograms of the ‘lena’ image after encryption for the three channels respectively on the right side. They depict that the encryption does not leave any concentration of a single pixel value.

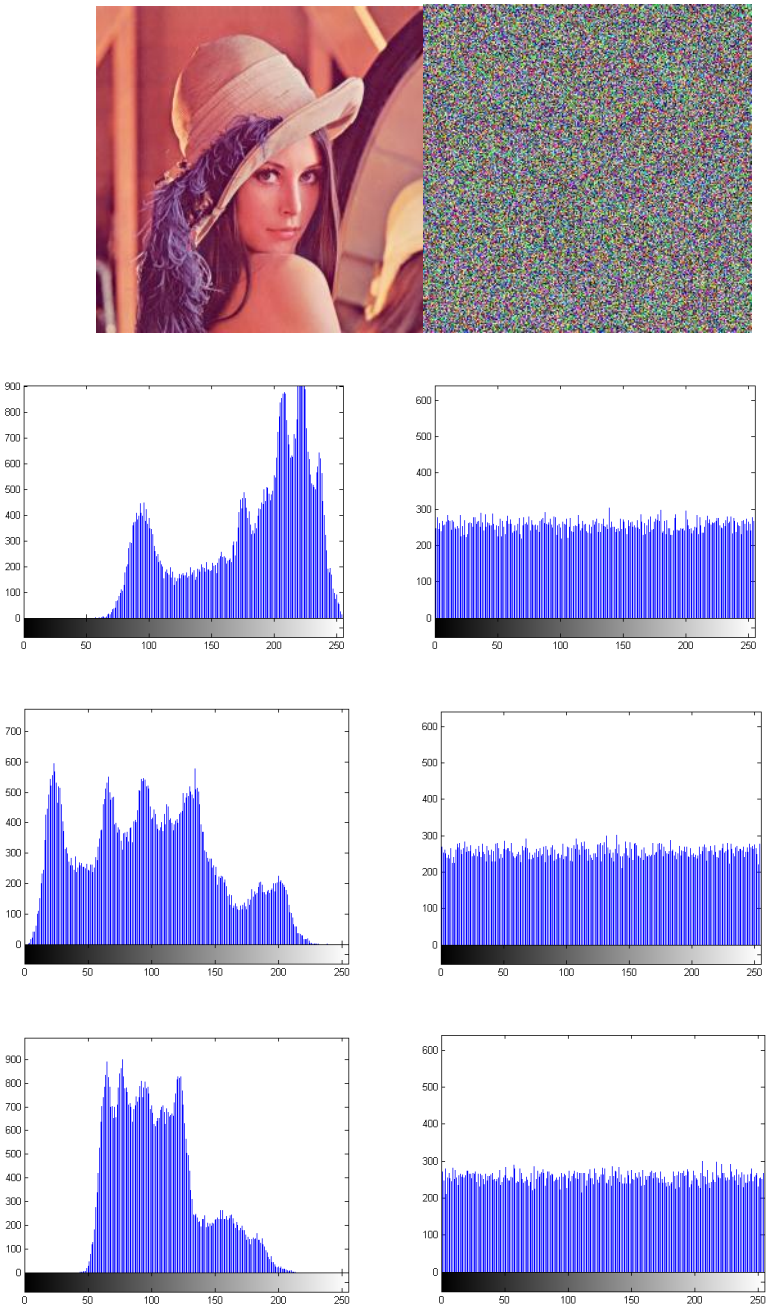

Figure 5.Left Side: Histogram of 'lena' plain image for red, green and blue channels (top to down). Right Side: Histogram of encrypted 'lena' image for red, green and blue channels (top to down).

4.1.2 Correlation of Adjacent Pixels

In a plain image the adjacent pixels show a high degree of correlation in horizontal, vertical and diagonal directions. The encrypted image should have a very small degree of correlation among its adjacent pixels. We select 1000 random pairs of pixels from an image and the following formula gives the correlation coefficient.

$$corr_{xy} = \frac{C(x,y)}{\sqrt{D(x)D(y)}} \quad (2)$$

where,

$$C(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \quad (3)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2 \quad (4)$$

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

Here x_i and y_i form the pair of i^{th} adjacent pixels and N is the total number of pairs.

Table 1 shows the correlation coefficient values of the six plain images (Fig. 4) between horizontal, vertical and diagonal adjacent pixels. It can be noted that the adjacent pixels are highly correlated.

Table 1.Correlation Values of Plain-Images

Channels	Plain Images	Horizontal	Vertical	Diagonal
RED	Lena	0.9558	0.9781	0.9336
	Bridge	0.8680	0.9070	0.8287
	Lake	0.9234	0.9201	0.8886
	Mandrill	0.8474	0.8032	0.7944
	Peppers	0.9371	0.9392	0.9077
	Plane	0.9205	0.9092	0.8546
GREEN	Lena	0.9401	0.9695	0.9180
	Bridge	0.9055	0.9131	0.8700
	Lake	0.9354	0.9272	0.8943
	Mandrill	0.7285	0.6674	0.6487
	Peppers	0.9657	0.9673	0.9451
	Plane	0.8938	0.9174	0.8419
BLUE	Lena	0.9189	0.9495	0.8948
	Bridge	0.9354	0.9411	0.9138
	Lake	0.9377	0.9401	0.9099
	Mandrill	0.8030	0.7914	0.7625
	Peppers	0.9259	0.9330	0.8928
	Plane	0.9179	0.8912	0.8563

Table 2 shows the correlation coefficient values for the Red, Green and Blue channel of the cipher images formed by encrypting the plain images with the proposed encryption algorithm. The cipher images bear very little resemblance to the original images and that the adjacent pixels in the horizontal, vertical and diagonal directions are correlated to a very small degree.

Table 2.Correlation Values of Cipher-Images

Channels	Plain Images	Horizontal	Vertical	Diagonal
RED	Lena	-0.0014	-0.0012	0.0004
	Bridge	-0.0040	-0.0066	-0.0010
	Lake	-0.0052	-0.0011	0.0018
	Mandrill	0.0034	0.0001	0.0033
	Peppers	-0.0014	-0.0034	-0.0016
	Plane	-0.0024	-0.0043	0.0088
GREEN	Lena	0.0004	0.0067	-0.0026
	Bridge	-0.0053	-0.0017	0.0008
	Lake	0.0044	-0.0025	0.0068
	Mandrill	-0.0031	-0.0041	0.0029
	Peppers	0.0008	0.0027	0.0029
	Plane	0.0026	-0.0003	0.0014
BLUE	Lena	-0.0049	0.0014	-0.0005
	Bridge	0.0023	0.0001	0.0037
	Lake	-0.0010	-0.0044	0.0002
	Mandrill	0.0023	0.0001	-0.0014
	Peppers	-0.0016	-0.0006	0.0013
	Plane	0.0040	-0.0007	0.0041

4.1.3 Correlation between plain and cipher image

The previous section showed correlation between adjacent pixels of plain image or cipher image. But it is also necessary to have no relevant correlation between the plain image and the corresponding cipher image. Rather than using the pixel pairs of a single image, we use the pixels of the plain and cipher image at the same grid position.

The 2D correlation coefficients of the images are calculated by pairing the three channels of the plain image with the three channels of the cipher image. These form nine different pairs i.e. correlation between; red channel of plain image and red channel of cipher image, red channel of plain image and green channel of cipher image, red channel of plain image and blue

channel of cipher image; and so on for the green and blue channels of the plain image. These are represented as C_{RR}, C_{RG}, C_{RB}, C_{GR}, C_{GG}, C_{GB}, C_{BR}, C_{BG}, C_{BB}; where for any C_{ij}, i represents a channel (R,G,B) of plain image and j represents a channel (R,G,B) of cipher image. The coefficient values given in Table 3 depict that there is little or practically no correlation between the plain image and its corresponding cipher image. The cipher image thus displays characteristics of a random image.

Table 3.Correlation Values between Plain Image and Cipher Image

Images	C_{RR}	C_{RG}	C_{RB}	C_{GR}	C_{GG}	C_{GB}	C_{BR}	C_{BG}	C_{BB}
Lena	-0.0033	0.0016	0.0047	-0.0026	-0.0008	0.0006	-0.0029	0.0003	-0.0021
Bridge	-0.0029	0.0005	0.0003	-0.0020	-0.0006	0.0011	0.0008	0.0007	0.0010
Lake	-0.0012	0.0002	0.0005	-0.0041	-0.0007	0.0033	-0.0050	-0.0021	0.0039
Mandrill	-0.0019	-0.0004	-0.0024	-0.0035	0.0011	-0.0036	-0.0034	0.0005	-0.0036
Peppers	-0.0030	-0.0059	-0.0022	-0.0033	-0.0024	-0.0012	-0.0042	-0.0007	0.0005
Plane	0.0072	0.0014	-0.0003	0.0068	0.0025	0.0015	0.0057	0.0033	0.0033

4.2 Differential Analysis

Differential analysis displays the amount of change that the encryption performs on the image. The encryption of two very similar images should not have a similar distribution of pixels in the cipher image. In other words, cipher images of two plain images having just a single pixel difference, should not bear any pixel resemblance between them. An adversary should not be able to extract any meaningful relationship between plaintext and cipher text, by comparing the 2 different cipher text of similar plaintext.

NPCR (net pixel change rate) and UACI (unified average changing intensity) are used as measures of differential analysis. NPCR indicates the percentage of pixel change in the cipher image when a single pixel of plain image is changed. UACI measures the average intensity of the change between plain and cipher image.

Let us consider 2 cipher images X_1 and X_2, obtained by plain images P_1 and P_2 having difference of a single pixel. The pixel values at the grid position of i^{th} row and j^{th} column for the cipher images are denoted as $X_1(i,j)$ and $X_2(i,j)$. A bipolar array B is defined as follows

$$B(i,j) = \begin{cases} 0, & if\ X_1(i,j) = X_2(i,j) \\ 1, & if\ X_1(i,j) \neq X_2(i,j) \end{cases} \qquad (6)$$

Values for NPCR and UACI are calculated as given in equations (7) and (8), where W and H denote width and height of the cipher images, T denotes the largest supported pixel value in the cipher images (255 in our case) and abs() computes the absolute value. The NPCR and UACI values given in Table 4 show that the encryption algorithm is secure against differential attacks.

$$\text{NPCR} = \frac{\sum_{i,j} B(i,j)}{\text{W x H}} \text{x } 100\% \quad (7)$$

$$\text{UACI} = \frac{1}{\text{W x H}} \left[\sum_{i,j} \frac{abs(\text{x}_1(i,j) - \text{x}_2(i,j))}{T} \right] \text{x } 100\% \quad (8)$$

Table 4.NPCR and UACI Values Obtained for Encryption of 6 Plain images and Same Images with 1 Pixel Changed

Plain Images	NPCR	UACI
Lena	99.6333	33.4706
Bridge	99.5722	33.4403
Lake	99.5900	33.5313
Mandrill	99.6089	33.4595
Peppers	99.6185	33.4657
Plane	99.6206	33.4539

5. CONCLUSION

In this paper we proposed a new image encryption algorithm. The merits of the recent research, based on results, were combined along with a symmetric approach of encryption to provide a secure algorithm. The diffusion mechanism along with Feistel structure makes the algorithm stronger. The 3D Rossler system of equations is used for the random key generation. The splitting of the three dimensions of the key for the three channels makes the cryptanalysis to obtain the key more difficult. The experimentation performed depict that the algorithm generates favorable results.

REFERENCES

[1] Chang, C.-C., Hwang, M.-S.and Chen, T.-S., 2001. A New Encryption Algorithm for Image Cryptosystems. *Journal of Systems and Software*, Vol. 58, No. 2, pp. 83-91.

[2] Yano, K. and Tanaka, K., 2002. Image Encryption Scheme Based on a Truncated Baker Transformation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E85-A, No. 9, pp. 2025-2035.

[3] Gao, T. and Chen, Z., 2008. Image Encryption Based on a New Total Shuffling Algorithm.*Chaos, Solitons and Fractals*, Vol. 38, No. 1, pp. 213-220.

[4] Chen, G., Mao, Y. and Chui, C.K., 2004. A Symmetric Image Encryption Based on 3D Chaotic Cat Maps. *Chaos, Solitons and Fractals*, Vol. 21, pp. 749-761.

[5] Mao, Y., Chen, G. and Lian, S., 2004. A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps. *International Journal of Bifurcation and Chaos*, Vol. 14, No. 10, pp. 3613-3624.

[6] Guan, Z.-H., Huang, F. and Guan, W., 2005. Chaos Based Image Encryption Algorithm. *Physics Letters A*, Vol. 346, pp. 153-157.

[7] Zhang, L., Liao, X. and Wang, X., 2005. An Image Encryption Approach Based on Chaotic Maps. *Chaos, Solitons and Fractals*, Vol. 24, pp. 759-765.

[8] Gao, H., Zhang, Y., Liag, S. and Li, D., 2006. A New Chaotic Algorithm for Image Encryption. *Chaos, Solitons and Fractals*, Vol. 29, pp. 393-399.

[9] Pareek, N.K., Patidar, V. and Sud, K.K., 2006. Image Encryption Using Chaotic Logistic Map. *Image and Vision Computing*, Vol. 24, pp. 926-934.

[10] Wong, K.-W., Kwok, B.S.-H.and Law, W.-S., 2008. A Fast Image Encryption Scheme Based on Chaotic Standard Map. *Physics Letters A*, Vol. 372, pp. 2645-2652.

[11] Amin, M., Faragallah, O.S. and Abd El-Latif, A.A., 2010. A Chaotic Block Cipher Algorithm for Image Cryptosystems. *Communications in Nonlinear Science and Numerical Simulation*, Vol. 15, pp. 3484-3497.

[12] Patidar, V., Pareek, N.K. and Sud, K.K.,2009. A New Substitution-Diffusion Based Image Cipher Using Chaotic Standard and Logistic Maps. *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, pp. 3056-3075.

[13] Rossler, O.E., 1976. An Equation for Continuous Chaos. *Physics Letters A*, Vol. 57, No. 5, pp. 397-398.

[14] Kamat, V.G. and Sharma, M., 2014. Enhanced Chaotic Block Cipher Algorithm for Image Cryptosystems. *International Journal of Computer Science Engineering*, Vol. 3, No. 2, pp. 117-124.

This paper may be cited as:

Kamat V. G. and Sharma M., 2014. Symmetric Image Encryption Algorithm Using 3D Rossler System. *International Journal of Computer Science and Business Informatics, Vol. 14, No. 1, pp. 1-13.*