International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

A New Chaos Based Image Encryption and Decryption Using a Hash Function

Payal Verma[1], Prabhakar Sharma[2]

[1]*Research scholar Department of Computer Science and Engineering Raipur Institute of Technology, Raipur, India*
[2]*Assistant professor Department of Computer Science and Engineering Raipur Institute of Technology, Raipur, India*

Abstract **-** *Chaos-based image encryption/decryption techniques have recently been extensively studied due to their superior properties in efficiency and speed. However, many of the proposed schemes suffer from known plain-text attacks. This paper suggests a new, fast chaos-based image encryption and decryption with a plain-text related permutation. Permutation and diffusion is used for encryption process. To shuffle the position of image pixels and generate the diffusion key stream the Arnold's cat-map and Lorenz system is used, respectively. A hash function is used to generate permutation/diffusion key, in this murmur3 hash algorithm is employed to generate hash value. Hash functions are having the unique property that it will produce completely different shuffled images even if there is a tiny difference between images and it helps accelerate the diffusion process. As a result, the proposed scheme reduces the number of cipher cycles to achieve acceptable and good diffusion properties. Whereas, there are many number of cycles are required by previous schemes to achieve same properties. Thorough security tests are carried out with detailed analysis and the results demonstrate the high security of the new scheme.*

Key Words: image encryption, cat-map, Murmur hash, Lorenz system, image decryption

1. INTRODUCTION

Nowadays, a huge amount of digital images are being stored on different media and exchanged over internet and wireless network. It is very easy to disclose important information to illegal users. Therefore, encryption techniques are used to protect images from unauthorized data reading, alteration, addition or deletion. Encryption is the process which uses special mathematical algorithms and keys to transform digital data into cipher code before they are transmitted. Cipher code is not understandable by users. At the receiver side information is converted back to a understandable form using decryption process and consequently the information is conveyed securely. The main goal to encrypt the image is to provide authentication of users, integrity, accuracy, and security of data.

Digital images are mainly characterized by the bulk data capacity, high redundancy and strong correlation among adjacent pixels. Accordingly, modern block ciphers, such as DES, Triple-DES, AES and IDEA, whose keystreams are produced by key schedules or key expansion algorithms that work on integers, are not suitable for practical image encryption and decryption. Due to the intrinsic features of sensitive to the initial condition and system parameter, ergodicity and pseudo-randomness, the algorithm based on chaotic systems have shown promising results and high efficiency. The chaos based schemes produce keystreams by iterating chaotic systems or maps and quantifying their current values of state variables. Because of the chaotic systems or maps are performed on real number field and arithmetic operations, a direct way to improve the efficiency of chaos based image cipher is reducing the number of iterations required by key stream generator. Several methods have been suggested in accordance with this idea recently, and a brief overview is given below.

In [1], the selective image encryption using a spatiotemporal chaotic system is investigated. It concludes that only selectively encrypting 50% of the whole image data can gain acceptable security. Therefore, the encryption time is significantly reduced. In [2], defines efficient and improved diffusion approaches, which uses a simple table lookup and swapping techniques as a light weight replacement of the 1D chaotic map iteration. In [3], a fast encryption algorithm is proposed. It combines permutation and diffusion process. As a result, it needs only one scanning of image for combined permutation-diffusion and reduces the time of encryption. In [4, 5], it separates diffusion and permutation process. Chaotic map is used for diffusion and cat map is used for permutation process. Result indicate that it obtain satisfactory level of security with only one cipher cycle. In [6-8], schemes have introduced encryption process with using hash functions to generate key for encryption and decryption process. Hash function produces different cipher images for different images.

However, for most existing chaos based image ciphers, the diffusion key stream extracted from the chaotic orbit is solely determined by the key. The same key stream is used to encrypt different plain images unless a different key is used. Such cryptosystems can be easily cracked by using known or chosen plain text attack. To address this problem, a new chaos based image encryption and decryption using Arnold's cat map and Lorenz system is suggested in this paper. In permutation stage, plain image is shuffled by using cat map. Control parameters of cat map are given by the murmur3 hash value of the original image. As is known, the key property of hash function is that it produces completely different hash value for different messages or images. In

diffusion stage, a large key space is ensured as the state variables of Lorenz system are used as the diffusion key. As a result, the number of rounds is reduced to achieve an acceptable diffusion property and satisfactory level of security. Decryption is also performed by using same methodology. The rest of this paper is organized as follows. Section 2 discusses the permutation-diffusion strategies of proposed encryption and decryption scheme. Security of the proposed image scheme is thoroughly analysed in section 3. Finally, section 4 concludes the paper.

2. METHODOLOGY

2.1 Encryption process

The architecture for proposed encryption scheme is shown in fig.-1. Under this structure, two stages are performed i.e. permutation stage and diffusion stage. Firstly in permutation stage, the original image is passed to Arnold cat map to shuffle the pixel positions. The control parameters of cat map, also called the permutation key are given by the hash value of original image. There is an unique property of hash function is that it generates different hash values for different messages, i.e. it produces completely different shuffled images even if there is a slight change between original ones. Our scheme suggests the 32-bit version of murmurhash3 algorithm, proposed by Austin Appleby to produce the hash value. This algorithm has better performance than other ones because of its speed, essential behavior, and has better collision avoidance. In diffusion stage, a Lorenz system is employed to mask the shuffled data by generating a key stream. The detailed permutation and diffusion process are discussed as follows.

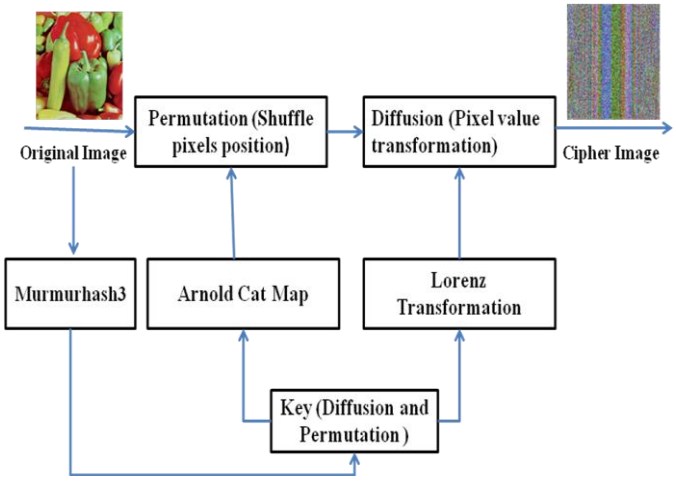


Fig -1: Architecture of the proposed encryption scheme

a) Permutation Process

In permutation image pixels are generally shuffled by an area preserving chaotic map, without change to their values. Arnold's cat map is used to achieve permutation.

Arnold Cat Map

Arnold's Cat Map was discovered by Vladimir Arnold in 1960. It apparently randomizes the original organization of plain image pixels. However, if iterated number of times, the original image reappears.

The Arnold cat map is a chaotic bijection of a unit square onto itself. This transformation of the image is obtained by implementing the following equation (1).

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \quad (1)$$

Where, p and q are positive integers i.e. control parameters, and N is the number of pixels in one row (column). The inverse transform of the map is found to be given by-

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \quad (2)$$

The value of control parameters are given by the 32-bit murmur3 hash value of plain image. First two bit of the murmur3hash value is passed to cat map as its control parameters. Utilization of small parameters also speeds up the calculation. The pseudo-code for murmur3 hash algorithm is listed below. From pseudo-code, bitwise and integer multiplication operations to manipulate and update the hash value is used by the algorithm. By the use of these operations the computational cost of the hash algorithm is much lower than that of one round of diffusion process, where the real numbers are manipulated.

When Arnold cat map algorithm is executed once, the coordinates of original pixel positions will be transferred from the (x_n, y_n) to a new position (x_{n+1}, y_{n+1}), then the process is repeated with the matrix multiplied. Process will iterated continuously. The number of iteration is T and the original image size is N and, p and q is correlated. Thus, whenever the value changes, it generates a completely different shuffled image for every iteration.

Pseudo-code of MurmurHash3 Algorithm

```
Murmur3_32 (key, len, seed)

  c1 ← 0xcc9e2d51
  c2 ← 0x1b873593
  r1 ← 15
  r2 ← 13
  m ← 5
  n ← 0xe6546b64
  hash ← seed
  for each fourByteChunk of key
     k ← fourByteChunk

     k ← k × c1
     k ← (k ROL r1)
```

```
    k ← k × c2

    hash ← hash XOR k
    hash ← (hash ROL r2)
    hash ← hash × m + n

 with any remainingBytesInKey
    remainingBytes ←
SwapEndianOrderOf(remainingBytesInKey)
  remainingBytes ← remainingBytes × c1
  remainingBytes ← (remainingBytes ROL r1)
  remainingBytes ← remainingBytes × c2

    hash ← hash XOR remainingBytes

  hash ← hash XOR len

  hash ← hash XOR (hash SHR 16)
  hash ← hash × 0x85ebca6b
  hash ← hash XOR (hash SRH 13)
  hash ← hash × 0xc2b2ae35
  hash ← hash XOR (hash SHR 16)
```

b) Diffusion Process

In 1963, Edward Lorenz, an early developer of a chaos theory, established a simplified mathematical model for atmospheric convection. Three ordinary differential equations are recognized as the Lorenz equation, is described by-

$$\begin{cases} x' = \sigma(y - x), \\ y' = x(\rho - z) - y, \\ z' = xy - \beta z, \end{cases} \quad (3)$$

Where σ, ρ, β are the system parameters. When σ=10, ρ=8/3, β=28, the system display chaotic behavior. The value of three initial state variables x_0, y_0, and z_0 are used as the diffusion key. Compared with 1D chaotic maps such as logistic map, tent map, the Lorenz system is more complicated dynamical property and number of state variables. Therefore, the cryptosystem based on Lorenz system has stronger unpredictability and larger key space, which are suitable for secure cipher.

The detail diffusion process is described as follows:
Step 1: The pixels of shuffled image are arranged to a vector $P = \{p_0, p_1, \ldots p_{N\times N-1}\}$ in the order from left to right, top to bottom.

Step 2: Make a key stream with length equal to *p*. The following steps are used to generate key stream.

Step 2.1: To avoid the harmful effect of transitional procedure pre-iterate system (3) for I_0 times. The forth order Runge-Kutta method is employed to solve the equation.

Step 2.2: Lorenz system is iterated continuously. Iterate system (3) for *t* times, where $t = (N\times N/3)$. For each iteration, we can get three state values and one is selected as quantification of key stream element.

Step 2.3: a key stream $k = \{k_0, K2 \ldots k_{N\times N-1}\}$ is qualified according to following equation-

$$k_n = mod[sig\_n(abs(s_n)), 2^L] \quad (4)$$

Where *L* is color depth of plain image, *sig_n(x)* returns the *n* most significant decimal digits of *x*, and *abs*(*x*) returns the absolute value of *x*. All variables are declared as double-precision type in our scheme which has a precision of 15 decimal digits.

Step 3: calculate the cipher pixels value according to following equation-

$$c_n = k_n \oplus \{[p_n + k_n] mod\ 2^L\} \oplus c_{n-1} \quad (5)$$

Where c_n, p_n, k_n, c_{n-1} are the output cipher pixel, currently operated plain pixel, key stream element, and previous cipher pixel respectively, and $\oplus$ performs bit-wise exclusive OR operation. One may set the initial value c_0 as a constant.

Step 4: Return to step 2 until all shuffled image pixels are encrypted in order from left to right and top to bottom.

As seen from equation (5), the modification of particular pixel is not only depends on the corresponding key stream element, but also it depends on the effect of all the precious pixel values. As a result, the influence of a single plain pixel can be spread out over many cipher pixels. It makes the cryptosystem more robust against differential attack.

2.2 Decryption Process

The architecture of decryption scheme is shown in fig.-2. Decryption process is exactly opposite to proposed encryption process. In this the same secret key is used because the proposed scheme is a symmetric method. The steps for decryption process are discussed below.

Due to the fact that the proposed scheme is a symmetric scheme, in the decryption process the same secret key is used, which leads to the same key stream $k = \{k_0, k_1, \ldots k_{N\times N-1}\}$ for Lorenz system of decryption process. Next two steps represent the decryption process.

Step 1: On the pixels of the encrypted image $C = \{c_0, c_1 \ldots c_{N\times N-1}\}$, we apply the diffusion process to obtain the auxiliary image $A = \{a_0, a_1 \ldots a_{N\times N-1}\}$. All steps for diffusion process are same as encryption process except step 3:

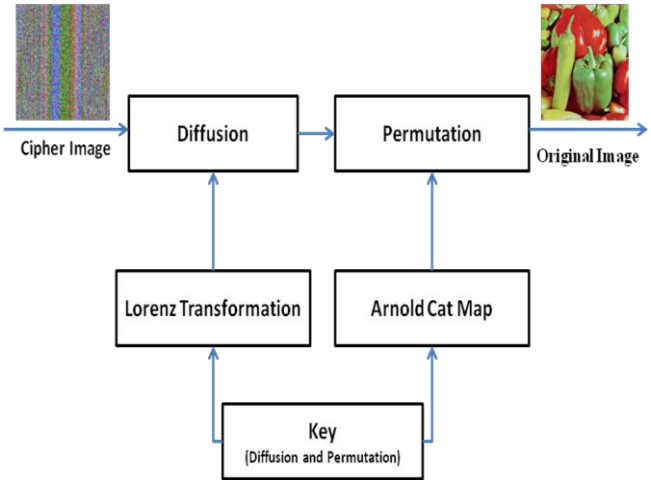


Fig -2: Architecture of the proposed decryption scheme

Calculate the auxiliary pixel value according to equation (6)-

$$a_n = [k_n \oplus c_n \oplus c_{n-1} + 2^L - k_n] mod\ 2^L \quad (6)$$

Where c_n, a_n, k_n, c_{n-1} are the input cipher pixel, output auxiliary pixel, key stream element, and previous cipher pixel respectively, and $\oplus$ performs bit-wise exclusive OR operation.

Step 2: On the pixels of auxiliary image $A = \{a_0, a_1 \ldots a_{N\times N-1}\}$, we apply the inverse transformation of the cat map (1) for permutation.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \quad (7)$$

This inverse transformation is used to obtaining the plain image $P = \{p_0, p_1 \ldots p_{N\times N-1}\}$.

3. SECURITY ANALYSIS

In this section, for the proposed image encryption/decryption system we performed thorough security tests which are carried out with detailed analysis to demonstrate the high security of the proposed scheme. All tests are preformed for each of three color channels (red, green, and blue) in order to achieve detailed analysis of proposed scheme.

3.1 Key Space Analysis

An effective cryptosystem should have enough large key space to resist against brute-force attacks. The key of the proposed cryptosystem having two parts- permutation key (*key-P*) and diffusion key (*key-D*). Permutation key is given by the control parameters (*p, q*) of the Arnold cat map and the diffusion key is given by the initial conditions (x_0, y_0, z_0) of the Lorenz system. Both are independent to each other.

The space of permutation key is N^2 because of control parameters are range from 1 to *N*. the Lorenz system is declared as double precision type, the space of diffusion key is approximately $(10^{15})^3$. Therefore, the total key space is-

$$Key\text{-}S = key\text{-}P\ (N) \times key\text{-}D \quad (8)$$

Consider an image of size 256×256 as example; the total size is approximately $256^2 \times 10^{45} \approx 2^{165}$. Generally cryptosystems with a key space greater than 2^{100} are considered to be "computational security". Therefore, our proposed scheme has large enough key space to resist differential attacks.

3.2 Statistical Analysis

a) Histogram Analysis

An image histogram is a graphical representation showing a visual analysis of the distribution of pixels by plotting the number of pixels at each color channel. It is the most often used visual impression tool to study the distribution of a color image of pixel values frequencies are plotted separately for each color channel. The histogram of a peppers test image (fig.3 (a)) and its encrypted image (fig.3 (c)) produces by proposed scheme are shown in fig.3 (b) and fig.3 (d) respectively.

From fig.3, one can see that the histogram of ciphered image is significantly different from that of original image. It is clear that the pixels in cipher image are perfectly uniformly distributed, and hence does not provide any clue to employ statistical analysis.

b) Entropy Analysis

The distribution of pixel values can be further qualitatively determined by calculating the information entropy of the image. The information entropy is usually expressed by the average number of bits needed to store and communicate on symbol in a message. The entropy H(*X*) of an image source *X* can be calculated as-

$$H(X) = -\sum_{i=1}^{N} P(x_i) log_2 P(x_i), \quad (9)$$

Where *X* is a random variable with *N* outcomes $\{x_1 \ldots x_N\}$ and $P(x_i)$ is the probability mass function of outcome x_i from a color channel RGB if an image and the entropy is expressed in bits. For instance, for a ciphered image with 256 color image, the entropy should ideally be $H(X) = 8$, otherwise there exist certain degree of predictability which threatens its security. The information entropy of some test images and their corresponding cipher images produced by the proposed scheme are calculated, and their results are listed in table.1. from table.1, the entropy of all the output cipher images are between 7.99881 and 7.99951, which are very close to maximal theoretical value 8. This means the proposed scheme produces output with perfect randomness.

Table.-1. Results of information entropy analysis.

Test Image Name	pepper	House	Jally beans	Fruits	Serano
Plain	7.6887	7.5181	6.8527	7.6475	7.2903
Cipher	7.9991	7.9991	7.9992	7.9988	7.9991

c) Correlation of Adjacent Pixels

In an ordinary image, each pixel is highly correlated with its adjacent pixels either in horizontal, vertical, or diagonal directions which is indicated by a value of Pearson's correlation coefficient very closes to 1. However, an efficient image cryptosystem should produce the cipher image with sufficiently low correlation in the adjacent pixels. Fig.4 (a) and (b) show the correlation distribution of two horizontally adjacent pixels in peppers test image (fig.3 (a)) and its output cipher image (fig.3 (b)). In fig.4 we plotted the value of the pixel at position (*x, y*) and the value of the pixel at position (*x+1, y*) from the peppers image. We repeated the same plotting for vertically adjacent pixels (fig.4 (c) and (d)), respectively, for diagonally adjacent pixels (fig. 4(e) and (f)).

The visual testing of the correlations between adjacent pixels can be done by the following procedure. First, randomly select 3000 pairs of adjacent pixels in horizontal, vertical, and diagonal direction from the plain/encrypted image, and, using the values from each RGB color channel. Then, we calculate the correlation coefficient r_{xy} of each pair using following formula:

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2)(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2)}} \quad (10)$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (11)$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (12)$$

Where x_i and y_i are values of ith pair of adjacent pixels, and N denotes the total number of samples. Table 2 listed the Calculated correlation coefficients for adjacent pixels in peppers test image and its output cipher image for RGB color channel. From the table it can be seen that the correlation coefficients for adjacent pixels in the output cipher image are very close to zero, and it further confirms that the encryption process eliminates the inherent strong existing correlation between the pixels of the plain image. This fact proves that the proposed system will resist against attacks of statistical type.

2.3 Differential Attacks Analysis

To implement differential attack, an opponent usually makes a slight change in the plain image and observes the changes of corresponding cipher image to find out some meaningful relationship between plain image and cipher image. Testing the security of a cryptosystem against differential attacks is necessary to evaluate how a minor change in original image is reflected upon the cipher image. Two common criteria are used to measure the diffusion capacity of an image encryption scheme: number of pixel change rate (*NPCR*) and unified average changing intensity (*UACI*).
Consider two images, whose corresponding plain images and ciphered images; be denoted by P_1 and P_2. A bipolar array, D with same size as image P_1 and P_2 is defined. Then, $D(i, j)$ is determined by $P_1(i, j)$ and $P_2(i, j)$. The NPCR is used to measure the percentage of different pixel numbers between the plain image and cipher image.

NPCR is defined as-

$$NPCR = \sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j) \times \frac{100\%}{M \times N} \quad (13)$$

Where,

$$D(i,j) = \begin{cases} 0, if\ P_1(i,j) = P_2(i,j) \\ 1, if\ P_1(i,j) \neq P_2(i,j) \end{cases} \quad (14)$$

(a)

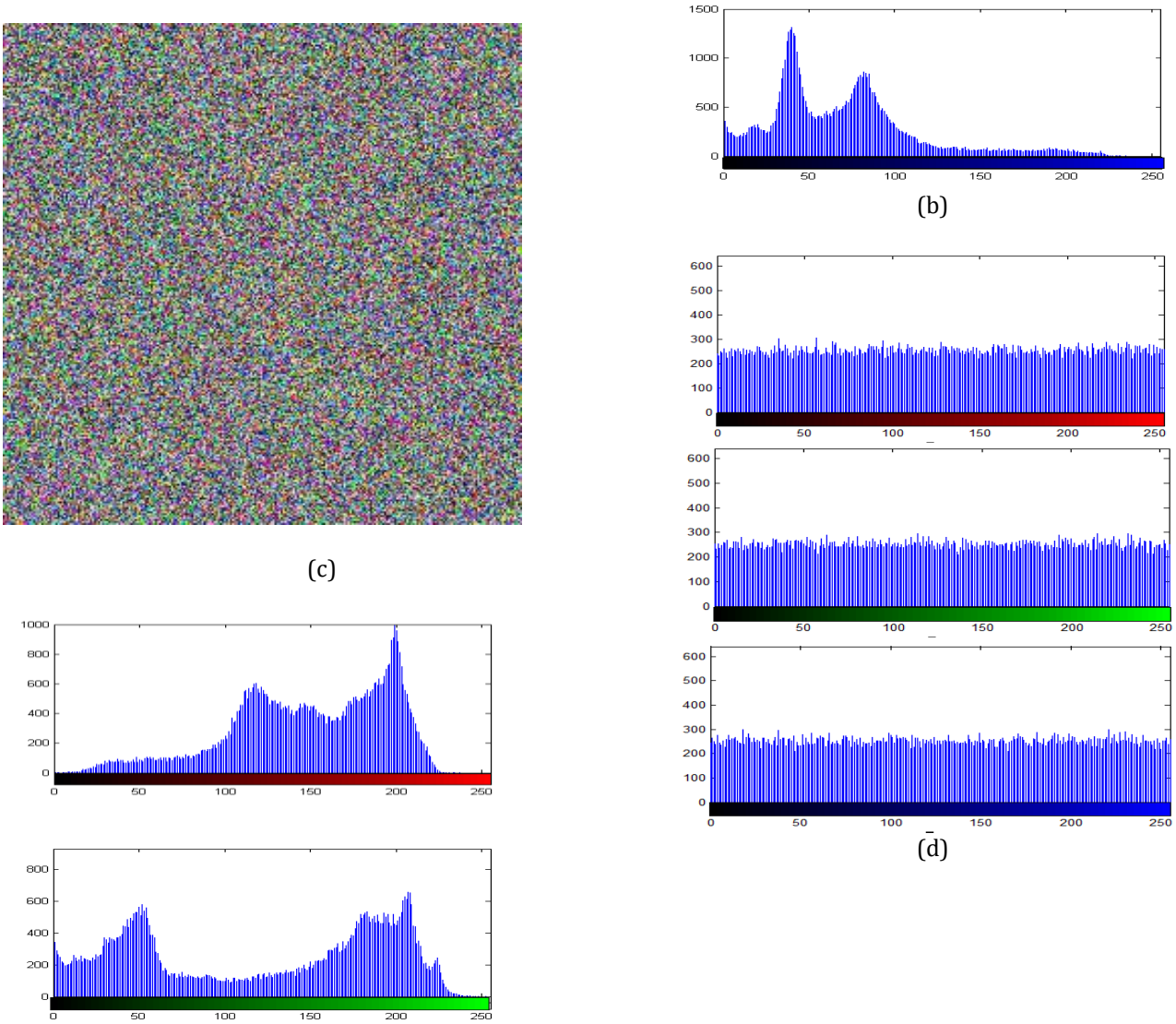


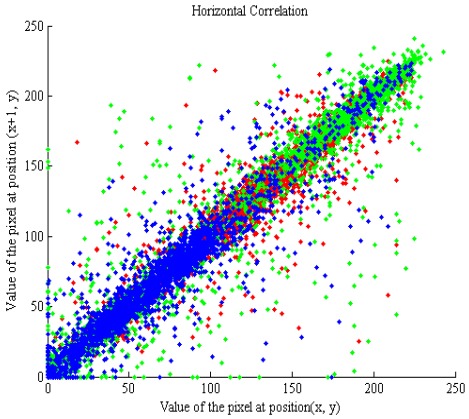
Fig -3 Histogram of test image (a) the peppers test image, (b) histogram of (a), (c) the ciphered image of (a), (d) histogram of (c)

UACI is used to measure the average intensity of differences between the two images. It is defined as-

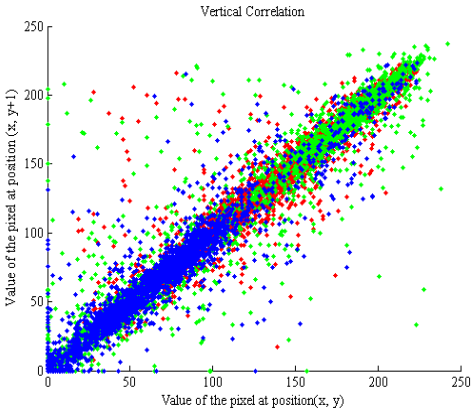
$$UACI = \left[\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|P_1(i,j)-P_2(i,j)|}{255}\right]\times\frac{100\%}{M\times N} \quad (15)$$

The *NPCR* and *UACI* values for two truly random images with 256×256 color image, namely the expected values for a good cryptosystem, are 99.609% and 33.464%, respectively [9].

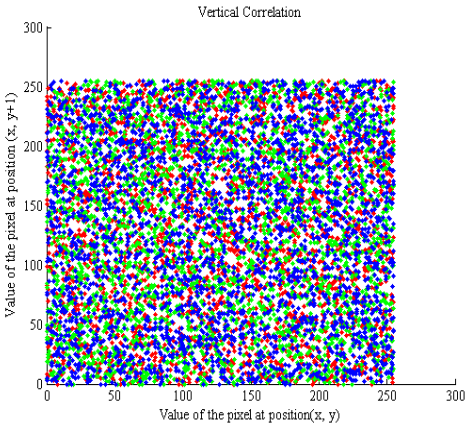
To evaluate the *NPCR* and *UACI* of the proposed cryptosystem, we assume a worst case that two plain images have only one pixel difference, as shown in fig.5 (a)



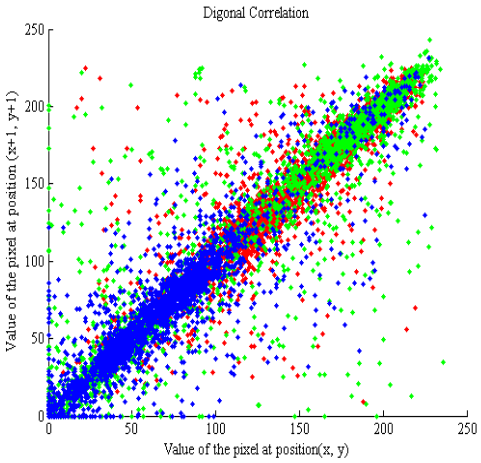
(a) Correlation of horizontally adjacent pixels in the original image



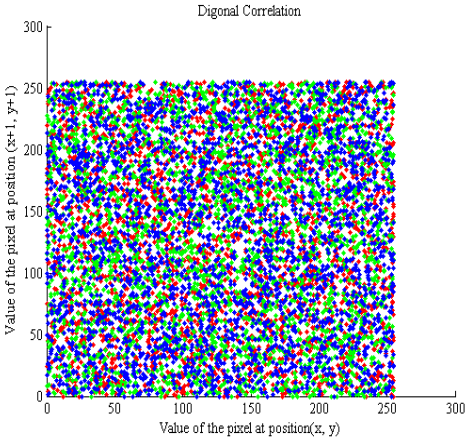
(c) Correlation of vertically adjacent pixels in the original image



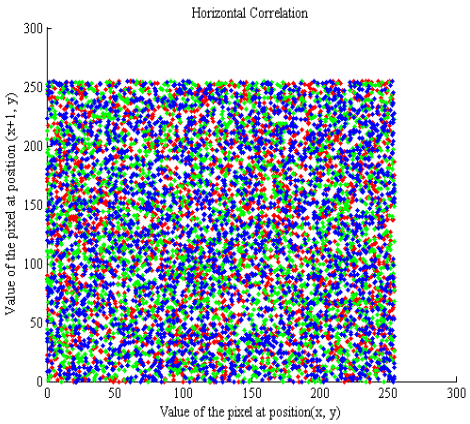
(d) Correlation of vertically adjacent pixels in the ciphered image



(e) Correlation of diagonally adjacent pixels in the original image



(f) Correlation of diagonally adjacent pixels in the ciphered image



(b) Correlation of horizontally adjacent pixels in the ciphered image

Fig.4- Correlation of adjacent pixels from peppers plain/encrypted image.

and (b). Their corresponding cipher images are shown in fig.5(c) and (d), respectively. The differential image between the two cipher images can be found in fig.5 (e).

We obtain *NPCR* = 99.621% and *UACI* = 33.465% for peppers test image we also test some such pairs of images, the results are compared with that of pervious scheme [8],

as listed in table.3. From the table, the proposed scheme reduces the cipher cycles to achieve an acceptable diffusion property, whereas many number of cipher cycles are needed by previous scheme to achieve the same properties. Therefore, the proposed scheme has a superior computational efficiency.

Table.2- correlation coefficients for adjacent pixels in some test images and their output cipher images.

Test Image Name		Original			Ciphered		
		Horizontal	vertical	Diagonal	Horizontal	vertical	Diagonal
Peppers	Red	0.9495	0.9575	0.9372	-0.0087	-0.0127	0.0174
	Green	0.9372	0.9426	0.9604	-0.0150	0.0062	0.0178
	Blue	0.8996	0.9270	0.8980	0.0126	-0.0074	0.0165

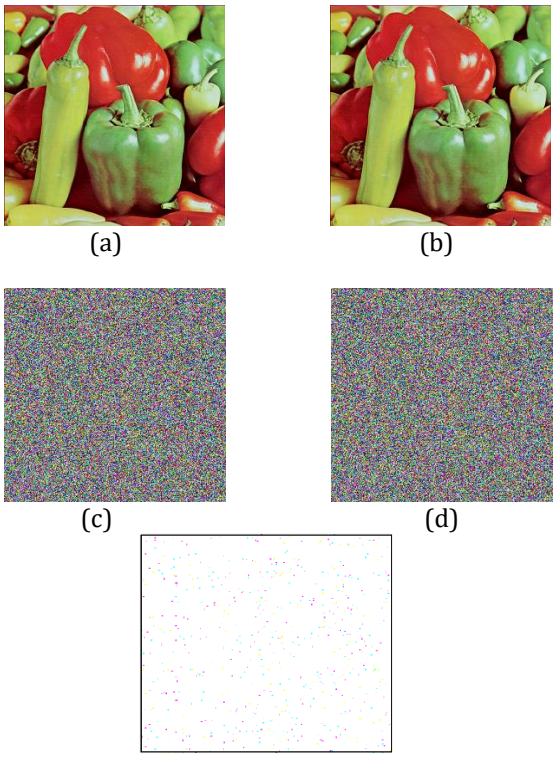


Fig.5- diffusion capacity test, (a) and (b) are two plain images with only one pixel difference, (c) cipher image of (a), (d) cipher image of (b), (e) differential image between (c) and (d).

2.4 Quality of Decryption Process

Within the cryptosystem performances evaluation, the quality of the decryption should be also checked. Basically, this testing is consists to analyze that the image obtained after decryption process is exactly same as plain image. In this sense, we evaluate the mean squared error (*MSE*) between the plain image $P = \{p_0, p_1 \ldots p_{n-1}\}$ and the corresponding decrypted image $D = \{d_0, d_1 \ldots d_{n-1}\}$, on each RGB color channel, using the following formula:

Table.3- Result of NPCR and UACI tests.

Images	Round 1		Round 2	
	NPCR_1	UACI_1	NPCR_2	UACI_2
Peppers	0.9960	0.3346	0.9962	0.3341
House	0.9962	0.3342	0.9961	0.3345
Lena	0.9977	0.3082	0.9996	0.3370
Jellybeans	0.9962	0.3345	0.9962	0.3344
Fruits	0.9959	0.3342	0.9965	0.3345
Serrano	0.9960	0.3340	0.9960	0.3349

$$MSE(P,D) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left(P(i,j)-D(i,j)\right)^2}{M\times N} \qquad (16)$$

A value close to 0 of MSE indicates a good quality of the decryption process [10], while other value indicates the occurrence of error in this process. In all tests performed using proposed scheme, the value of MSE was 0 for each RGB color channel, which indicates that decryption is carried out without any loss of information.

4. CONCLUSIONS

Development of new chaotic maps which meet the current demands of security is an actual research direction in the field of chaotic cryptosystem. The main objective is to obtain a large key space, induced by control parameter and initial conditions, for which the map is chaotic and ergodic.

In this paper, we have proposed a new and fast chaos based image encryption and decryption scheme with a permutation-diffusion strategy. In permutation stage the cat map is used to shuffle the position of pixels. Murmur3 hash value of original image is calculated to determine the control parameters of the cat map. In the diffusion stage, the Lorenz system is used to generate the diffusion key. Decryption is also performed by using the same key and the process is exactly opposite to proposed encryption scheme. Experimental results indicate that the proposed scheme reduces the number of cipher cycles to achieve an acceptable and a satisfactory level of security. Compared with previous scheme that requires many number of cipher cycles to achieve the same security level. It implies that the proposed cryptosystem has stronger unpredictability and larger key space. Thorough security analysis has been carried out with detailed numerical analysis, key space analysis, statistical analysis, and differential analysis, which has demonstrated the satisfactory security of new scheme and gives a batter speed performance as compared to previous schemes.

5. REFERANCES

[1] T. Xiang, K. W. Wong, and X. Liao, "Selective image encryption using a spatiotemporal chaotic system," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 17, no. 2, article no. 023115, 2007.

[2] K. W. Wong, B. S. H. Kwok, and C. H. Yuen, "An efficient diffusion approach for chaos-based image encryption," Chaos, Solitons and Fractals, vol. 41, no. 5, pp. 2652–2663, 2009..

[3] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," Appl. Soft Comput., vol. 11, no. 1, pp. 514–522, 2011.

[4] S. El Assad and M. Farajallah, "A new chaos-based image encryption system," Signal Process. Image Commun., vol. 41, no. March, pp. 144–157, 2016.

[5] J. Zhang and J. Wang, "A chaos-based digital image cryptosystem with an improved diffusion strategy," Lect. Notes Electr. Eng., vol. 270 LNEE, no. VOL. 1, pp. 763–770, 2012.

[6] M. Farajallah, Z. Fawaz, S. El Assad, and O. Deforges, "Efficient image encryption and authentication scheme based on chaotic sequences," Secur. 2013 - 7th Int. Conf. Emerg. Secur. Information, Syst. Technol., no. c, pp. 150–155, 2013.

[7] Y. Wang, D. Xiao, and Y. Wang, "One-way hash function construction based on 2D coupled map lattices on 2D coupled map lattices," no. March, 2007.

[8] C. Fu, O. Bian, H. Y. Jiang, L. H. Ge, and H. F. Ma, "A new chaos-based image cipher using a hash function," 2016 IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. ICIS 2016 - Proc., pp. 0–5, 2016.

[9] Y. Wu, S. Member, J. P. Noonan, and L. Member, "NPCR and UACI Randomness Tests for Image Encryption," 2011.

[10] R. Boriga, A. Cristina, and A. Diaconu, "A New One-Dimensional Chaotic Map and Its Use in a Novel Real-Time Image Encryption Scheme," vol. 2014, 2014.