Image Encryption Using Chaos Maps

B. Sai Kumar[1], L. Vikhyath[2], R. Geetha Krishna Pavansai[3]

[1,2,3]Student, Sreenidhi Institute of Science and Technology, Ghatkesar, Telangana

Abstract— As the exchange of data over the open networks and Internet is rapidly growing, security of the data becomes a major concern. The solution is to encrypt the data. The data can be text, image, audio, video etc. Images are one of the largest media in this generation. Earlier image encryption techniques like Advanced Encryption Standard (AES), Data Encryption Standard (DES), Rivest–Shamir–Adleman (RSA) etc. exhibit low levels of security and weak anti attack ability. This problem was overcome by using chaos-based cryptography. The chaotic systems are very sensitive to initial conditions and control parameters which make them suitable for image encryption. The main aim of this project is to examine the efficiency of different chaotic maps, Chaos technology encrypts images such that the possibility of deciphering is reduced greatly as the cipher text presents a randomness. Chaos based encryption is a very important means of modern digital encryption. The statistical characteristic if the original image is transformed, thereby increasing the difficulty of unauthorized breaking of the encryption. Various performance metrics like peak signal to noise ratio (PSNR), mean-square error (MSE) and execution time evaluation are performed on the obtained decrypted image.

Index Terms— Chaos Maps, Henon Map, Arnold Cat Map, Logistic Map, Image Encryption, PSNR, MSE, Autocorrelation.

1 Introduction

Over the past two decades, the world has witnessed tremendous advancements in the area of multimedia-based applications like medical imaging, and multimedia image/video database services. The rapid growth of digital media and multimedia-based applications has increased the requirement of transmitting them over public networks. The key obstacle in such applications for scientific and research community has been the efficient and secure transmission over the network. A digital image is either a gray image represented as two-dimensional matrix having pixel intensities or a colored image represented as three-dimensional matrix or RGB matrix where the three planes of matrix correspond to red, green and blue respectively. Since images are bulky in data, therefore encryption algorithms should be designed in such a way that are faster, involve fewer complex computations and secure to various possible attacks.

The search for new encryption schemes suitable to image/ video data, improvements to existing cryptographic techniques and proofs of security continues at a rapid pace. A number of encryption algorithms suitable for image data has been suggested in literature. Chaos is one such dimension widely used in such algorithms. The phenomenon of chaos theory was first introduced by Edward Lorenz in 1972 with conceptualization of “Butterfly Effect". There is a link between chaos features and traditional cryptography, according to many academics.

According to Q.V. Lawande(2005), the strength of cryptography is found in the selection of strong keys, which are confidential parameters utilized in encryption. Selection of strong keys makes it difficult for the cryptanalyst to guess the key.

Chaotic systems are highly sensitive. So, if the starting circumstances or control parameters are chosen as "Keys" and "Pathways" for encryption/decryption, chaotic maps can be beneficial in encryption/decryption schemes. Chaos-based encryption systems are symmetric since they employ the same parameters for encrypting and decrypting. In chaotic maps, ergodicity and sensitivity to beginning circumstances show excellent confusion and diffusion qualities, which are critical for an efficient encryption strategy. Further, the control parameters and initial conditions of chaotic maps form a very large key space enhancing the security against brute force attack.

2 Literature Survey

One of the most essential ways for ensuring the security of digital photographs and videos transmitted over the internet is encryption. Because of the inherent characteristics of image/video, such as mass data volume, strong correlation between neighboring pixels, and high data redundancy, encryption methods that can handle the underlying main challenges of data storage, speed, and security are required. Various real-time image/video encryption approaches have been proposed in the literature, which are faster and more secure than traditional encryption algorithms. One of the paradigms that appears to be useful for cryptography is chaos. Various encryption strategies have been proposed in the literature that use chaos to increase the security of an encryption scheme due to its properties.

2.3 Motivations of the Proposed Research Work

Traditional encryption algorithms such as DES (Data Encryption Standard), RSA (named after Rivest, Shamir, Adleman) & AES (Advanced Encryption Standard) are focused on number theory and insist spatial information, and are based on image/intrinsic video's characteristics, e.g. massive data capacity, strong correlating between adjacent pixels and high data redundancy. For multimedia-based encryption techniques, data storage, speed and security are the key issues.

As seen from literature survey, a lot of research work has been carried out in these directions but there are still possibilities of making image encryption process more efficient and robust. Motivations for this research work are listed below-

Chaotic values are useful in generating keys, hash functions, digital signatures etc. which are useful in cryptography. Use of chaos has made the encryption process comparatively faster because of ease to generate long chaotic pseudorandom sequences whose values seems to be uncorrelated if initial value and related parameters for generation of sequence are not known but still security is a big concern and an efficient encryption technique robust to attacks remains an ongoing challenge in research community. The work proposes an efficient chaos-based encryption technique that is analyzed in conjunction with different security parameters like statistical analysis, key-space analysis, analytics of key sensitivity, entropy analysis, sound and block loss and other encryption schemes.

3 Methodology

3.1 Henon Map Image Encryption Algorithm

The chaotic map of Henon is used as a symmetric chipboard system. The Henon chaotic map creates pseudo-random binary code, which was described below, two dimensional non-linear dynamic time

$$X_{n+1} = 1+Y_n - aX_n^2$$
$$Y_{n+1} = bX_n \quad n=0, 1, 2\ldots \qquad (3.1)$$

The parameters a and b are of parameters because the system's dynamic behavior depends on these values. The system can still be unpredictable only if the values of a and b are 1.4 and 0.3. Turbulent, discontinuous, or periodic orbits. The starting points X1 and Y1 serve as symmetric keys of the chaotic cryptosystem used for encryption on the sender side and decryption on the receiver side. Since the Hénon map is deterministic, the decoding of the encoded image reconstructs the original image on the receiver side at the same point of origin. X1 and Y1. Therefore, the encryption and key sensitivity algorithms work together to prevent all types of code-breaking attacks.

$$X_{n+1} = 1+Y_n - 1.4X_n^{\,2}$$
$$Y_{n+1} = 0.3X_n \qquad (3.2)$$

3.1.1 Image Encryption by Henon Chaotic System

The shuffled image is encrypted using pseudo-random binary sequence generated by taking key values for Henon map.

Step 1: Choose the starting value of (X1, Y1) on the Henon card. This value serves as the initial secret symmetric key for the Hénon card.

Step 2: The Hénon card acts as a key stream generator for cryptographic systems. The dimensions of the sequence is the size of the image and the number of henon sequence is obtained through equation (3.2) when the size of the image is m/n.

Step 3: The empirical analysis concluded that the threshold of 0.3992 was determined to balance the series. Then, as shown in equation (3), the decimal value is converted to a binary value based on this threshold value. Where Z is a binary sequence.

$$Z_i = \begin{cases} 0 \; if\; Xi \leq 0.3992 \\ 1 \; if\; Xi \geq 0.3992 \end{cases} \qquad (3.3)$$

Step 4: The Henon pattern will be reduced to a decimal value by integrating all eight straight bits.

Step 5: Encryption is performed using an OR operation that excludes every bit between the composite image and the sequence generated in step 4.

3.1.2 Decryption of Encrypted Image

The chaotic behavior of the system is deterministic, so it finally reconstructs the image using the same keys (X1, Y1). How to decode a mixed image. This composite image is placed in the exact reverse order of encoding, and the original image is captured by the receiver.

3.2 Arnold Cat Map Image Algorithm

Chaos is a common technology used in the generator of random numbers, because this algorithm is more simple and can be used both for storage and process objects in the process stream bound objects, only a few functions (chaotic maps) and some (first conditions) are used quite well if the process takes a considerable length of time.

Arnold's Cat Map is a chaotic two-dimensional image that can be applied to alter the image pixels position without discarding any information from the image; S = { (x, y)∣ x, y = 0, 1, 2... N-1}. N. The following equation can be used to write the 2-D image of the Arnold Cat Map:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} (\bmod n)$$

$$\begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\bmod n) \qquad (4.1)$$

When p and q are positive, the determinant (A) = 1. (X' y'), when Arnold's cat mapping algorithm is performed, this is the new location of the original pixel (x, y). From the cat map of Arnold to the number of iterations, R represents a random drawing containing in the original image all values for the same pixel. The number of iterations for R to carry out varies according to p, q and N factors. The cat mapping algorithm of Arnold however has the p, q and R iterations, all of which can be used as a secret key. Arnold's algorithm for cat mapping:

Step 1: Get image matrix and gray scaled image matrix.

Step 2: Arnold Cat encryption

- Read image
- Iterate until it reaches key
- Call Arnold Cat transform method

Step 3: Arnold Cat transform

- R([x,y]) = [(x + y) mod n, (x + 2y) mod n] (4.2)

Obtained R matrix is the required map that is used in encryption and decryption.

Step 4: Arnold Cat Map decryption

- The number of iterations 'n' at which the original image will reappear is given by these rules of thumb:
- Here 'd' is the dimension of the square image:

1. if d = 2(5^i) for i >=1, n = 3*d
2. if d = (5^i) for i >=1, n = 2*d
3. if d = 6(5^i) for i>=1, n = 2*d
4. else n <= 12*d / 7 (4.3)

Where d is the decrypted image matrix

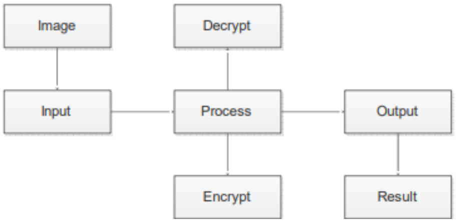


Figure 4.1 Decryption an image flow diagram

The below is an example to illustrate Arnold cat map encryption:

104	134	144	139	150	163	162	155	161
88	124	140	139	152	167	159	146	135
84	123	141	140	154	169	157	138	136
81	120	135	124	133	152	162	164	150
78	121	141	132	135	149	159	162	156
76	123	148	139	136	143	149	154	153
79	123	148	141	137	139	138	139	138
83	120	141	137	136	136	130	128	125
82	116	134	131	132	133	129	129	127
77	111	130	127	128	131	134	142	138
70	107	129	126	124	130	141	156	146

As a first test, we consider the RGB values of 3x3 pixels for the values shown in the Table, and its outcomes.

Pos Pixel	0	1	2
0	104	134	144
1	88	124	140
2	84	123	141

The next encryption and decryption algorithms are done after the pixel value has been obtained. The next procedure is Arnold's Cat Map The process of encryption and decryption involves pixel wiggle and pixel hop as follows:

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} \bmod 3 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0 & 0\\ 84 & 219\end{bmatrix} = \begin{bmatrix}0\\303\end{bmatrix} \bmod 3 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}0\\2\end{bmatrix} = \begin{bmatrix}0 & 0\\ 168 & 438\end{bmatrix} = \begin{bmatrix}0\\606\end{bmatrix} \bmod 3 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1 & 134\\ 0 & 0\end{bmatrix} = \begin{bmatrix}135\\0\end{bmatrix} \bmod 3 = \begin{bmatrix}2\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1 & 134\\ 84 & 303\end{bmatrix} = \begin{bmatrix}135\\387\end{bmatrix} \bmod 3 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}1 & 134\\ 168 & 438\end{bmatrix} = \begin{bmatrix}135\\606\end{bmatrix} \bmod 3 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}2\\0\end{bmatrix} = \begin{bmatrix}2 & 268\\ 0 & 0\end{bmatrix} = \begin{bmatrix}536\\0\end{bmatrix} \bmod 3 = \begin{bmatrix}2\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}2 & 268\\ 84 & 303\end{bmatrix} = \begin{bmatrix}536\\387\end{bmatrix} \bmod 3 = \begin{bmatrix}2\\0\end{bmatrix}$$

$$\begin{bmatrix}1 & 134\\ 84 & 134+84+1\end{bmatrix} x \begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}2 & 268\\ 168 & 438\end{bmatrix} = \begin{bmatrix}536\\606\end{bmatrix} \bmod 3 = \begin{bmatrix}2\\0\end{bmatrix}$$

This process represents an iterative procedure – for the process of iteration, first, the number of pixels depending on the user's wishes is versatile, the author uses one iteration only to demonstrate the Cats map algorithm and, after this iteration, the results are the following pixels:

Pos Pixel	0	1	2
0	84	134	104
1	88	124	140
2	144	123	141

3.3 Logistic Map Image Encryption Algorithm

Process of step-by-step image encryption and decryption using two chaotic logistic maps.

1. An 80-bit exterior private key is used in the preferred image encryption procedure. The private key is also split into 8-bit blocks, each of which is referred to as a session key.

$$K = k_1k_2.k_{20} \text{ (hexadecimal); (5.1)}$$

The alphabetic characters (0–9 & A–F) are the ki's here, and the session key is represented in each group of two alphanumeric characters. The secret key can also be displayed as an ASCII mode

$$K = K_1K_2.K_{10} \text{ (ASCII); (5.2)}$$

Every Ki here is an 8-bit secret key block, i.e. session key.

2. Two chaotic logistic maps in the proposed algorithm have been used to achieve the following objective of image encryption:

$$X_{n+1} = 3{:}9999X_n(1-X_n);\ (5.3)$$

$$Y_{n+1} = 3{:}9999Y_n(1-Y_n):\ (5.4)$$

Consistent maintaining the system parameter values in the algorithm of the two logistic maps (3.999). That is a very chaos case, but for those maps the original conditions (X0 and Y0) are calculated with a certain amount of data. Session key operation.

3. To calculate the initial condition X0 for the first logistics map, select three blocks of the session key. Convert to binary string like, K4K5K6:

$$B1 = K_{41}K_{42}\ldots K_{48}K_{51}K_{52}\ldots K_{58}K_{61}K_{62}\ldots\ldots K_{68};\ (5.5)$$

Here, the k_{ij}s are the session key's binary numbers (0 or 1). We then use the above binary representation to calculate a real number X01:

$$X01 = (K41x2^0 + K42x2^1 + \ldots\ldots + K_{48}x2^7 + K_{51}x2^8 + K_{52}X2^9 + \ldots + K_{58}X2^{15} + K_{61}X2^{16} + K_{62}X2^{17} + \ldots.. + K_{68}X2^{23})/2^{24}. \qquad (5.6)$$

Further, we compute another real number X_{02} as follows:

$$X_{02} = \sum_{i=13}^{18} (k_i)_{10}/96, \qquad (5.7)$$

Here, as explained in Eq, ki's are secret key elements in hexadecimal mode. [1]. We are now calculating the initial X0 condition for the first X01 and X02 logistic map.

$$X_0 = (X_{01} + X_{02}) \bmod 1 \qquad (5.8)$$

4. Repeating the first logistic diagram using the initial conditions obtained in step 3, we generate a sequence of 24 real numbers f1, f2. .., f24. Other values will be removed from the string. A sequence of real numbers is converted to a sequence of whole

numbers according to the following formula:

$$P_k = \text{int}(23 \times (f_k - 0.1)/0.8) + 1,$$

where $k = 1,2,\ldots,24.$

5. We choose three blocks of session keys, i.e. K1K2K3, and convert it into a binary string in order to calculate initial condition Y0 for a second logistic map:
$$B_2 = K_{11}K_{12}\ldots K_{18}K_{21}K_{22}\ldots K_{28}K_{31}K_{32}\ldots K_{38},$$

Here k_{ij}'s are the session key i th block binary numbers (0 or 1). We then use the above binary representation to calculate the true Y01 number:

$$Y_{01} = (B_2)_{10}/2^{24}. \qquad (5.9)$$

In addition, we calculate the following real number Y02:

$$Y_{02} = \left(\sum_{k=1}^{24} \underline{B_2[P_k]} \times 2^{k-1}\right)/2^{24}, \qquad (5.10)$$

B2[Pk] in this case denotes that the Pkth bit value is either 0 or 1. In the B2 binary string, i.e. Now we compute the primary state Y0 for the second logistic map using Y01 and Y02 as:

Y0 = (Y_{01} + Y_{02}) mod 1: (13)

6. Read three bytes consistently from the image file. These three bytes are red, green, and blue (RGB) values and form one pixel of the picture together.

7. Range [0,1,0.9] Place in 24 intersecting spaces and in 8 various groups. Then allocate to each of these groups different types of activities. Table 1 presents every operation's groups and time range.

Once again, use the first condition Y0 of step 5, reiterate the second logistic map. The result of the second logistic map determines what the red, green and blue (RGB) bytes are to be encrypted / decrypted. This step is (K10) 10 (i. Finally, the red, green and blue bytes are entered in the file (decimal number corresponding to the key of the 10th session).That is, a single pixel encoding is performed.

For the next 15 pixel image file repeat the steps 6 and 7

8. After encrypting a 16-pixel block of the image file, session keys K1 through K9 are modified as follows:

$$(K_i)_{10} = ((K_i)_{10} + (K_{10})_{10}) \bmod 256, \quad (1 \le i \le 9). \qquad (5.11)$$

The first logistic map (i.e., X0 = f24) after changing the private key as above and getting the final value of X in step 4 as the conditional initial value is also explained in step 4. Generate a sequence of 24 real numbers as described above, and repeat steps 5 to 7 until all the image files are exhausted.

This is very similar to the encryption method described above. The only difference is at step 7. That is, for the cryptographic process, the second logistics label (K10) is repeated 10 times, after which the corresponding cryptographic operation (depending on the result of the logistics card) is performed. The second logistic map (K10) is iterated 10 times initially for the decryption process, then the appropriate decryption operation is performed (in reverse order again, depending on the outcomes of the following maps).

4 Analysis

The analysis of these algorithms is done by plotting Intensity histograms and adjacent pixel autocorrelation graphs.

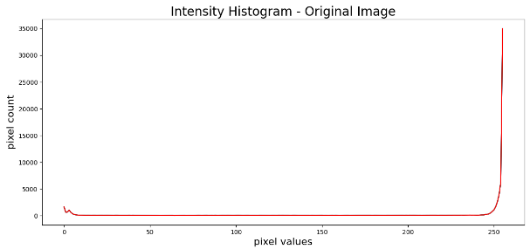
4.1 Intensity Histogram

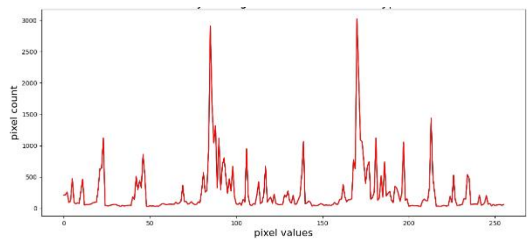
The histograph of the picture generally refers to the pixel intensity value histogram in the context of the image processing. This histogram is a histogram showing the number of pixels for each of the distinct picture intensity values. Grayscale pictures may be 256 distinct intensities, such that the histogram depicts 256 pixel allocation numbers graphically between these values of the grayscale. The histogram of a colored picture is also available. You can get separate histograms for the red, green, and blue channels, or you can create a 3D histogram showing the red, blue, and green channels and their respective brightnesses on three axes. Score represents the number of pixels. The operation is subject to implementation. This might be an image of the chart in a suitable picture format, or a data file describing the chart's stats.

It's really simple to operate. The image was examined in a single scan, retaining and generating the suitable histogram the amount of consecutive pixels that was identified at each intensity value. Three algorithms show changes in pixel intensity on a histogram.

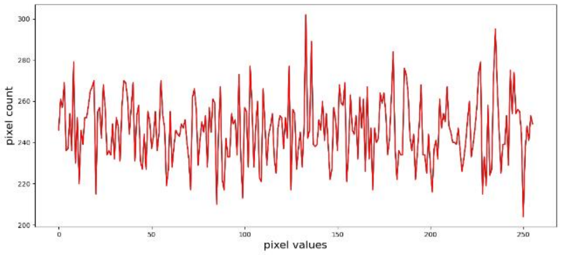
The Intensity histograms of original, encrypted images for all 3 algorithms shows the variation of pixel intensity through a graph.

The image treatment describes a digital image's tone distribution using an intensity histogram. The observer can evaluate the full tonal distributed on a glance by glancing at the histogram of the given image. The horizontal axis indicates the change in the colour and the vertical axis shows the number of pixels.





Intensity Histogram of Encrypted image (Henon Map)



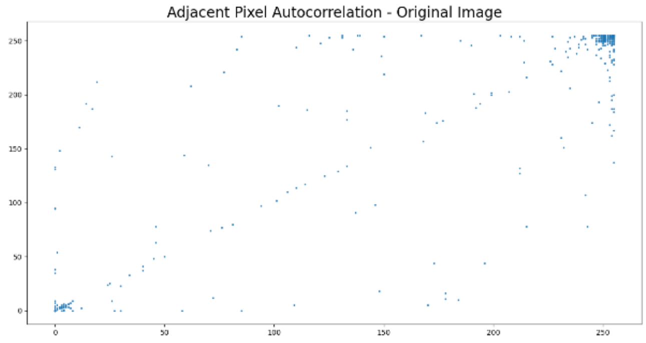
Intensity Histogram for Encrypted image (Logistics Map)

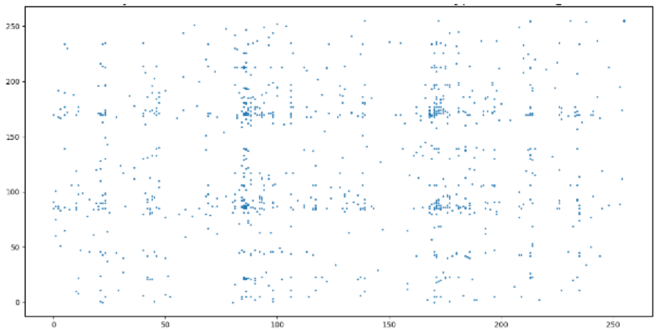
4.2 Adjacent Pixel Autocorrelation

As new image encryption and encryption algorithms are developed, researchers are using a variety of methods to evaluate the effectiveness and security of the algorithms. For example, global Shannon entropy measurement, histogram analysis, differential cryptographic analysis, and neighbor pixel correlation. The image encoding/encoding algorithms will produce an output image that is significantly (i.e., statistically) different from the pure version. In this context, the main limitation of traditional assessment methods is that they provide qualitative rather than qualitative measurements. As a result, many researchers have come together in recent years to improve traditional assessment methods and develop new statistical tests for the randomness of images.

Modified during encoding and uses histogram analysis to describe the distribution of pixels of a single image relative to the encoded image. Histogram analysis is generally limited to visual assessment, i.e., whether a uniform distribution of the histogram of the encoded picture is considerably different from the picture of the origin. However, a more detailed analysis evaluates the relevance of the histogram. That is, we use the chi-square test to evaluate it as a qualitative measure of the distribution of histogram values in a coded image that approaches the characteristics of a uniform distribution.

The digital image coding scheme, for coded images, makes these correlations as close as possible to the zero values of the individual images. The same set of adjacent pixels is well distributed in the histogram.





Adjacent pixel Autocorrelation of Encrypted Image

5 Conclusion

In this project we saw execution and working of three main Chaos Algorithms namely Arnold map, logistic and henon map. The conclusion that we came to after this project is that:

- Arnold cat map takes more time to compute when compared to other two algorithms i.e., the run time of this map is comparatively very high.
- Logistic map generates key based on the map function and this key is used for further encryption.
- Henon map has less execution time than that of both the other algorithms i.e., it is faster.

After doing performance measures we came to a conclusion that:

- Logistic map has the least PSNR value among them.
- Arnold cat map encryption has the highest PSNR.

The conclusion here is that depending on the application in which we need the encryption we need to choose our map. If we need highest PSNR and Highest Encryption we need to go for Arnold cat map but the catch here is we need to compromise our run time.

If execution time is more important than that of level of encryption, we can choose Logistic map Encryption.

If we need to use encryption in an application which needs moderate level of encryption and faster execution time than that of Arnold cat map encryption we can use Henon map encryption.

6 Future Scope

As we are now aware of how to encrypt images, we can move forward to encrypt videos i.e., by encrypting each frame of the video. We can store the data to the cloud directly that is encrypted. Implementing encryption for cloud data increases security. We can move forward to encrypting using neural networking techniques which will help us reduce time complexity of the code and also increase the encryption rate. Using parallel computing technology to see the execution of our algorithm i.e., encryption in one computer and decryption in another computer.

7 References

[1] Monjul Saikia, Vikash Beruah (2017) Chaotic Map Based Image Encryption in Spatial Domain: A Brief Survey. In: Mandal J., Satapathy S., Sanyal M., Proceedings of the First International Conference on Intelligent Computing and Communication. Advances in Intelligent Systems and Computing, vol 458. Springer.

[2] K. Sakthidasan, B. Krishna, "*A New Chaotic Algorithm for Image Encryption and Decryption of Digital Color Images*", International Journal of Information and Education Technology, vol. 1, no. 2, June 2011

[3] Nitu Hazarika, Manjul Saikia, "*A Novel Partial Image Encryption Using Logistic Map*", 2014 International Conference on Signal Processing & Integrated Networks (SPIN), IEEE, 2014

[4] S. Kromodimoeljo, Teori & Aplikasi kriptografi, Medan, Indonesia: SPK Consulting, 2009.

[5] R. Munir, Pengolahan Citra Digital, Jakarta, Indonesia: Informatika , 2004.

[6] N. A. Abbas, "*Image encryption based on Independent Component Analysis and Arnold's Cat Map*", Egyptian Informatics Journal, pp. 139-146, 2016.

[7] R. Purba, A. Halim and I. Syahputra, "*Enkripsi Citra Digital Menggunakan Arnold's Cat Map Dan Nonlinear Chaotic Algorithm*", JSM STMIK Mikroskil, vol. XV, no. 2, pp. 61-71, 2014.

[8] E. AVAROĞLU, "*Pseudo Random Number Generator Based on Arnold's Cat Map and Statistical Analysis*", Turkey, 2011.

[9] Devaramapati Amarnath Gupta, Singh and I. Mangal, "*Image Encryption using Arnold Cat Map and S-Box*", IJARCSSE, vol. IV, no. 8, pp. 807-812, 2014

[10] Chen, Jun-xin & Zhu, Zhi-liang & Fu, Chong & Yu, Hai & Zhang, Li-bo. (2015). A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. Communications in Nonlinear Science and Numerical Simulation. 20. 846–860. 10.1016/j.cnsns.2014.06.032.

[11] Somaya Al-Maadeed, Afnan Al-Ali, Turki Abdalla, "A New Chaos-Based Image-Encryption and Compression Algorithm", Journal of Electrical and Computer Engineering, vol. 2012, Article ID 179693, 11 pages, 2012.

[12] Shenyong Xiao, YaShuang Deng, "Design and Analysis of Chaos-Based Image Encryption Algorithm via Switch Control Mechanism", Security and Communication Networks, vol. 2020, Article ID 7913061, 12 pages, 2020.

[13] Abhinav Sure, Tanuj Kalgutkar and Supeksha Waghmare. “Chaotic maps Image Encryption and Decryption.” (2012).

[14] Chaos Cryptography: Theory, Algorithms and Applications. Germany: Springer Berlin, 2011.

[15] Gao, Tiegan & Chun, Zenqiang. (2008). Image encryption based on shuffling algorithm. Chaos, Solitons & Fractals. 38.213-220. 10.1016/j.chaos.2006.11.009.