New 2D CA based Image Encryption Scheme and a novel Non-Parametric Test for Pixel Randomness

BalaSuyambu J
Department of Mathematics, IIT Madras, Chennai, Tamil Nadu-600036, India
ma11d002@smail.iitm.ac.in

Radha R
Department of Mathematics, IIT Madras, Chennai, Tamil Nadu-600036, India
radharam@iitm.ac.in

Rama R
Department of Mathematics, IIT Madras, Chennai, Tamil Nadu-600036, India
ramar@iitm.ac.in

Abstract

In this paper we have proposed a new test for pixel randomness using non-parametric method in statistics. In order to validate this new non-parametric test we have designed an encryption scheme based on 2D cellular automata. The strength of the designed encryption scheme is first assessed by standard methods for security analysis and the pixel randomness is then determined by the newly proposed non-parametric method.

Keywords: 2D Cellular Automata, Image Encryption, Pixel Randomness, Non-Parametric test.

1 Introduction

Multimedia images are widely used in internet communications, so the need for securely transmitting these images over networks has become an essential part of the field of data security. Image cryptography plays a vital role in securing confidential images. The purpose of image cryptography is to hide the content of the images by encrypting them so as to make the images unrecognizable to the intruders. One part of this paper deals with a new image cryptosystem using 2D Cellular Automata.

In general, there are two different methods to protect an image; they are (i) image shuffling and (ii) image encryption. Pixels positions are rearranged in image shuffling whereas in image encryption, pixel values and positions are changed. In both the cases it is very essential to check the security of
John von Neumann proposed a new emerging concept called Cellular Automata (CA)\cite{12}. CA is a discrete model consisting of regular grid of cells, each in one of the finite number of states. According to some fixed rule, the state of each cell will be changed in terms of the state of the current cell and the states of the cells in its neighborhood. Like 1D, higher dimension CA also can be defined\cite{8}. It is already proved that some of the rules of CA will be able to generate complex random patterns\cite{14}. In the last two decades 1D and 2D CAs are used in Cryptography. Lot of research is going on for CA based image cryptography \cite{1,2,10}. The paper is organized as follows: Section 2 describes the basics of 2D cellular automata and the 2D CA concepts used for our encryption scheme. Section 3 presents our new non-parametric test for pixel randomness. Section 4 describes our 2D CA based encryption scheme. Section 5 presents the simulation results and performance evaluations. Section 6 gives the concluding remarks.

## 2 2D Cellular Automata

The extensional behavior of 1D Cellular Automata is also able to produce complicated patterns in two-dimension. This extension is significant since this is compared with pattern formation in physical systems. 2D CA is a regular 2D lattice of cells. Each cell has n possible values and is updated in each discrete time steps according to a rule f that depends on the value of sites in some neighbor around it. There are different types of lattices and neighborhood structures in a 2D CA. Figure 1 shows the two familiar neighborhood structures named as von-Neumann neighborhood and Moore neighborhood. The value of $a_{ij}$ of a cell at $(i,j)$th position in $(t+1)$th time in a 2D CA with a rule f, that depends only on the cells according to von-Neumann neighborhood is evolved from $a_{ij}^{(t+1)} = f(a_{ij}^t, a_{i,j+1}^t, a_{i,j-1}^t, a_{i+1,j}^t, a_{i-1,j}^t)$.
2.1 Mathematical Model

1D and 2D CA are studied differently with the help of polynomial algebra and matrix algebra by researchers. P.P.Choudary et.al.already have studied and designed a new characteristic of 2D CA[4, 5]. These concepts are used for our proposed encryption method. 2D nine neighborhood Moores neighborhood is used here for evolutions. Different rules are defined in terms of dependencies from the following rule convention.

\[
\begin{array}{ccc}
64 & 128 & 256 \\
32 & 1 & 2 \\
16 & 8 & 4
\end{array}
\]

Here the central box is the cell being considered for evolution and the other boxes are its neighbors. The numbers in the boxes are used to specify the boxes for a particular rule. For example, rule 2 characterizes the dependency on its right neighbor whereas such dependency only on its bottom neighbor is characterized by rule 8 and so on. Rules, which are having dependency only on one cell are called fundamental rules. Suppose a cell has dependency on more than 2 neighbors. Then the rule number will be the sum of the numbers of the corresponding cells. For example 2D CA rule 15 (1+2+4+8) refers to the bottom-right 2x2 cells. There are 512 such rules including zero dependency. These 512 rules are classified into 9 different groups in terms of the number of 1s in the binary representation of the rule number. For example rule 8 belongs to Group 1, rule 320 belongs to Group 2 etc.,

2.2 Uniform CA Rules

Evolution of such 2D CA rules can be considered as matrix operations with 0s and 1s. If same rule is applied to each entry of the given problem matrix, then the CA is called uniform CA. Example 2.1 shows the application of the rule 15 in the problem matrix of size (2x3) with null boundary condition.

Example 2.1:

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\]

\[
\overset{\text{Rule 15}}{\rightarrow}
\begin{pmatrix}
0 & 1 & 0 \\
1 & 1 & 0
\end{pmatrix}
\]

Let \( S_t \) denote the binary matrix which represents the current state of a 2D CA. The next state of a cell is obtained by calculating XOR operation of the states in the corresponding boxes of a rule. In this example the states in the dependent neighbors of the 1st cell are 1,0,0,1 according to rule 15 and its XOR value 0 is its next state value.

To apply the rules to the problem matrix, one can use the rule matrices which are used in [9]. For a problem matrix of size \((m \times n)\), the size of the rule matrix \( R_n \) is \((mn \times mn)\) since the problem matrix is applied as a vector of order \((mn \times 1)\) got from the row major order.

Structure of rule matrices:
Let $R_n$ denote the rule matrix for rule $n$, for $n=0,1,2,...,511$. It is found that these rule matrices are related to each other in the following way. $R_1 \rightarrow$ Identity matrix. 

\[(R_2)^T = R_{32}, (R_4)^T = R_{64}, (R_8)^T = R_{128}, (R_{16})^T = R_{256},\]

where the power $T$ stands for the transpose of the matrix. By having these 5 fundamental matrices, one can easily get the rule matrices of all other rules using transpose and the XOR operations as given in example 2.2.

**Example 2.2:**

\[
\begin{align*}
R_{15} &= R_8 + R_4 + R_2 + R_1. \\
R_{82} &= R_{64} + R_{16} + R_2 = (R_4)^T + R_{16} + R_2. \\
R_{327} &= R_{256} + R_{64} + R_4 + R_2 + R_1 = (R_{16})^T + R_{64} + R_4 + R_2 + R_1. \\
\end{align*}
\]

In [3] P.P.Choudary et. al. have found new way of constructing rule matrices using the following two binary sequences $S_1$ and $S_2$.

$S_1 = \overbrace{111...10}^{(n-1) \text{'s}} \overbrace{111...10}^{(n-1) \text{'s}} \overbrace{111...10}^{(n-1) \text{'s}}$

That is $S_1$ starts with $(n-1)$ 1’s followed by one 0 then $(n-1)$ 1’s followed by one 0 and finally ends with $(n-1)$ 1’s.

$S_2 = \overbrace{0111...10111...10}^{(n-1) \text{'s}} \overbrace{0111...10}^{(n-1) \text{'s}}\overbrace{0111...10}^{(n-1) \text{'s}}$

That is $S_2$ starts with one 0 followed by $(n-1)$ 1’s and one 0 then $(n-1)$ 1’s and ends with $(n-1)$ 1’s followed by one 0, where $n$ is the number of columns in the problem matrix.

The five basic matrices are defined as follows with the structure of above sequences.

$R_1 = $ Identity matrix.

$R_2 = (n-1)$th diagonal elements are arranged like sequence $S_1$ and all other elements are 0.

$R_4 = (n+1)$th diagonal elements are arranged like sequence $S_1$ and all other elements are 0.

$R_8 = n$th diagonal elements are 1 and all other elements are 0.

$R_{16} = (n-1)$th diagonal elements are arranged like sequence $S_2$ and all other elements are 0.

Other rule matrices are defined as in example 2. Among all these 512 rules we are interested only in rules with non-singular matrices as they are reversible which are useful in Cryptography. There are only 31 rules that are reversible and those rules are expressed in terms of $R_1$ since they have 1’s in their diagonal.

### 2.3 Hybrid CA Rules

Unlike uniform CA, in hybrid CA each cell has its own local rule, that is, different rule is used for different cell. The total number of matrices of order $(mn \times mn)$ with entries 0 and 1 is $2^{(mn)^2}$, whereas we dealt with only 512 rules which are corresponding to linear Boolean rules. The rule matrices of hybrid rules we are discussing here are corresponding to remaining $2^{(mn)^2}-512$ matrices.
Example 2.3:

Let us consider the hybrid rule which defines as follows for a 2x2 problem matrix.
1. 1st cell changes its state by looking at the values in 1st, 3rd and 4th positions of the problem matrix.
2. 2nd cell changes its state by looking at the values in 2nd and 3rd positions of the problem matrix.
3. 3rd cell changes its state by looking at the values in the 4th position of the problem matrix.
4. 4th cell changes its state by looking at the values in the 1st, 2nd and 3rd position of the problem matrix.

$$\begin{pmatrix}
  a & b \\
  c & d
\end{pmatrix}$$

$$\begin{pmatrix}
  1 & 0 & 1 & 1 \\
  0 & 1 & 1 & 0 \\
  0 & 0 & 0 & 1 \\
  1 & 1 & 1 & 0
\end{pmatrix}$$

All such non-singular matrices of order \((mn \times mn)\) are reversible and they can be used for cryptography purposes.

3 A new Pixel Randomness Algorithm

Strength of an image encryption method measured by showing the randomness in the pixels of the shuffled and encrypted images. In literature we already have some parametric tests, which are used for checking randomness in the pixels of the images. But these tests are based on the assumption that the observations are treated as samples drawn from normal populations. If this is the case, these methods extract all the information that is available in a sample, and they usually attain the best possible precision. However, since there are many situations where it is doubtful whether the assumption of normality can be met, so the non-parametric tests came in to picture. It is useful to have a technique for testing whether a sample may be looked upon as random after it has been obtained. One such technique is based on the order in which the sample values were obtained, more specifically, it is based on the number of runs exhibited in the sample results[6].

A process which produces independent and identically distributed (i.i.d) samples is called a ”truly random process” in Statistics. A process is not considered truly random if the value in the sequence can be determined by its position. The property of iid is equivalent to the randomness, and it forms the basis for many statistical tests. Many practical applications will not have true random observations, but these simple tests are still useful to check how far the observations being close to randomness. It has very good application in cryptographic security, where it is used to check the randomness in various random number generators.

Investigations of randomness of a given sequence often require statistical tools for distribution comparison. Among them, goodness-of-fit tests and entropy estimates are two well-understood concepts[13]. However, when the distribution of the observed data is unknown, we have to go
for non-parametric tests, using some distribution invariant properties of random process. For example, the observations are transformed to a sequence of symbols that can specify their relative positions or magnitudes. The pattern of this sequence will serve as a measure of the randomness of the original process.

A run is defined as a succession of one or more identical symbols in the ordered symbol sequence. For example, the 0s and 1s in a bit string such as 000011111 and 0011001101 have 2 and 6 runs respectively. The number of runs and their sizes are used as a measure of the randomness of the observations. In a truly random sequence, we very rarely encounter too few, too many runs or runs with long length, etc., So the presence of any of these in an ordered symbol sequence is used as a statistical criteria for rejecting null hypothesis. Also these criteria are related to each other as too few runs results in some long length runs and too many runs results in some short length runs. So the number of runs is considered as a useful parameter in the randomness test.

Dichotomizing criteria is used for symbolizing the sequence for quantitative observations. By comparing each number in the sequence to a focal point (mean or median), it can be symbolized as + or depending on whether the number is greater or lesser than the focal point. Also, the trend or autocorrelation of the sequence can be determined from the relative magnitudes or ranks of adjacent numbers.

If a sequence contains $n_1$ symbols of one kind and $n_2$ of another kind, the sampling distribution of the total number of runs, $u$, can be approximated closely by a normal distribution with

$$
\mu_u = \frac{2n_1n_2}{n_1 + n_2} + 1 \quad \text{and} \quad \sigma_u = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1 + n_2)^2(n_1 + n_2 - 1)}}
$$

Thus the test of the null hypothesis $H_0$ that the arrangement of the symbols is random, can be based on the statistic

$$
Z = \frac{u - \mu_u}{\sigma_u}
$$

which has approximately the standard normal distribution.

In order to carry out the test for randomness, we set $H_0$ as arrangement is random against the alternating hypothesis $H_1$ as arrangement is not random. By setting the level of significance as $\alpha$, we make use of the following criterion for the critical region. If $z < -z_{\alpha/2}$ or $z > z_{\alpha/2}$ then reject $H_0$ and then we conclude the result as arrangement is random or not with $(1 - \alpha) \times 100\%$ level of confidence. We take $\alpha = 0.01$ and make the conclusion about randomness with 99% of confidence. One can see from the standard normal distribution table that $z_{\alpha/2} = 2.575$.

In our method the runs are formed in the following manner: First, $t$ distinct pixels are selected randomly from the image for which the pixel randomness is to be checked. Now for each of these $t$ pixels consider its Moore’s neighborhood pixels including the center pixel. Find the mean value of these 9 pixels for all the $t$ pixels. Now we have $t$ number of mean values. Now move horizontally (or vertically or diagonally) two steps from each of these $t$ pixels, again we have now $t$ points and find the $t$ number of Moore’s neighborhood mean values for these $t$ pixels. Now we have totally $2t$ mean values. Do the same process $h$ number of times, and we will have ‘$ht$’ mean values. Find the median
for all these ht mean values. Compare this median value with each of these ht mean values, if the mean value is less than the median then make the run entry as a otherwise b. Now we will have a sequence of a’s and b’s. This sequence is now exhibiting the runs. With these runs, the test has to be done using the equations 1 and 2. Moving along horizontal(or vertical or diagonal) direction is required to consider the distribution of pixels focally. The following algorithm explains the test.

**Algorithm:**

Notations:

- P is an image to be tested.
- N is the number of tests.
- T is the number of evaluations.
- n is the number of random pixels from P.
- h is the number of horizontal or vertical movements.

1. \( l = 1 \).
2. for \( i = 1 \) to \( T \)
3. \( c_1 = 0 \)
4. for \( j = 1 \) to \( N \)
5. Choose \( n \) random pixels \((p_1, p_2, p_3, .. p_n)\) without repetition.
6. for \( t = 1 \) to \( h \)
7. Compute \( n \) number of means \((m_{t1}, m_{t2}, m_{t3}, .. m_{tn})\) from each pixel’s Moore’s neighborhood.
8. Move 2 pixels horizontally(or vertically or diagonally).
9. end for
10. Find the median \( d \) for the \( hn \) means.
11. for \( s = 1 \) to \( h \)
12. for \( t = 1 \) to \( n \)
13. if \((m_{st} > d)\)
14. \( r_{st} = 1 \).
15. else
16. \( r_{st} = 0 \).
17. end if
18. end for
19. end for
20. Count the number of 1’s and 0’s in \( r \) and denote as \( n_1 \) and \( n_2 \) respectively.
21. Count the number of runs \( u \) in \( r \).
22. Compute \( \mu_u \) and \( \sigma_u \) using 1.
23. Perform the test using the statistic 2.
24. if \((-z_{\alpha/2} < z < z_{\alpha/2})\)
25. \( c_1 = c_1 + 1 \).
26. end if
27. end for
28. \( c_2[l] = c_1 \).
29. \( l = l + 1 \).
30. end for
31. \( C = (\sum_{k=1}^{T} c_2[k] / T) \times 100 \).

C measures the image pixel randomness in the range 1-100.
4 Encryption Scheme

The new 2D CA based encryption scheme in this paper is considered for gray scale and color images. Basic idea of the proposed image encryption, decryption method is based on the idea in [3]. Pixel values are changed by considering its binary representation using 2D CA. Encryption process has been done in two parts. In the first part, image scrambling is performed on the binary string form of the image and in the second part this scrambled image is been encrypted using 2D CA by considering the problem matrix of size (5 x 5).

4.1 Encryption Algorithm

This scheme can be applied on any square size problem matrix. We have implemented the scheme for the problem matrix of size (5 x 5). If the plain image size is mxn then the size of the plain image in binary form is mx8n. Pad rows and columns with 0s in the binary form such that the number of rows and columns are multiple of 5. Let the size of this matrix be (p1 x p2). Perform the permutation on the columns and rows of this matrix. This scrambled image is now considered as the input image for the 2D CA encryption scheme. Extract the (5 x 5) sub matrices in row major order from the scrambled binary image of size (p1 x p2). Now we have (p1 x 5 x p2) array P of (5 x 5) sub matrices. Consider each (5 x 5) sub matrix as a problem matrix and apply the proposed 2D CA scheme. Rules with invertible rule matrices are used for encryption purpose.

Encryption can be done in the following different ways:
W1: All (5 x 5) problem matrices in P use the same invertible uniform rule.
W2: Each row (column) of P uses different invertible uniform rules.
W3: Each problem matrix in P uses a different invertible uniform rule.
W4: All (5 x 5) problem matrices in P use the same invertible hybrid rule.
W5: Each row (column) of P uses different invertible hybrid rules.
W6: Each problem matrix in P uses a different invertible hybrid rule.

As we move from W1 to W6, the size of the key space is increased. After following any one from the above 6 ways, we will get the encrypted image. For the RGB color image, the encryption has to be done on each component separately. For the sake of increasing the difficulty of cryptanalysis, one can always follow W3 or W6 for encryption. The following algorithm explains the encryption process in W1.

Algorithm:
Get the matrix P from the binary scrambled plain image by clubbing (5x5) submatrices in row major order.

Input: Matrix P of size (mxn) of (5x5) submatrices.

Output: Matrix C of size (mxn) of size(5x5) submatrices.

1. Choose an invertible uniform rule r.
2. Construct the rule matrix R of size (25x25).
3. for i=1 to m
4. for j=1 to n
5. Get the vector P′(i, j) of size (25x1) from P(i, j) by ordering the elements in row major order.
6. C′(i, j) = R * P′(i, j). (where * is matrix multiplication in Z2).
7. Get the matrix C(i, j) of size (5x5) from C′(i, j) by arranging the elements in row major order.
8. end for
9. end for

We will get the encrypted image from the gray scale representation of the re-ordered binary matrix C.
4.2 Decryption

As we have used invertible uniform and hybrid rules for encryption, inverse exists for all the encryption rules. So the decryption can be done using the inverse rule matrices of the encryption rules matrices. For the color images, decryption can be done on the three components separately using the inverse rule matrix (matrices) of the corresponding encryption rule matrix (matrices).

5 Simulation Results and Performance evaluation

Several simulations are conducted to test various properties of the proposed 2D CA image encryption/decryption scheme including information concealing, confusion and diffusion properties. For the gray scale pepper image and the RGB color Lena images, we have performed all the ways of encryption which are mentioned above. Figure 2 shows the experimental results of the encryption scheme. (a) shows the gray scale Pepper image, (b),(c) and (d) show the encrypted images of (a) by W1, W2 and W3 respectively. (e) shows the RGB color Lena image, (f), (g) and (h) show the encrypted images of (e) by W4, W5 and W6 respectively.

![Figure 2: (a) Original Pepper Image; (b) Encrypted (a) by W1; (c) Encrypted (a) by W2; (d) Encrypted (a) by W3; (e) Original Lena Image; (f) Encrypted (e) by W4; (g) Encrypted (e) by W5; (h) Encrypted (e) by W6.](image)

5.1 Information concealing

If the key in the receiver side is identical with the secret key, the image can be revealed without loss. Figure 3 shows the results of our experiments with gray scale House image and the Lena color image, in both the experiments we could get back the original images using the correct secret key. In the figure 3, (a) and (d) are the original images, (b) and (e) are the encrypted images and (c) and (f) are the decrypted images.
5.2 Confusion Property

It is always essential to make the relationship between the key and cipher text as complex as possible so that the attackers can not get the key from the cipher text. Histograms of the original and encrypted images of the above experiments are given in Figure 4. The histograms of the encrypted images are almost uniform regardless of the original images and are significantly different from those of the original images. The results exhibit the confusion property of the proposed scheme.

5.3 Diffusion Property

In any strong cryptosystem, key plays the main role to spread out the redundant bits in the plain image over the cipher image along with the encryption method. So small changes in the key, produces a significantly different cipher image. This property is termed as diffusion property. We have performed encryption and decryption key sensitivity tests to show the diffusion property in our method. Figures 5 & 6 show the test results of encryption and decryption key sensitivity analysis respectively. These results clearly exhibit the diffusion property.

5.4 Statistical Analysis

5.4.1 Correlation Analysis

In a perfectly encrypted images and in a perfectly shuffled images, any pair of adjacent pixels along horizontal, vertical or diagonal directions should not be correlated. 1000 random pairs of adjacent pixels have been chosen horizontally, vertically and diagonally from the plain and encrypted images for the correlation analysis. The correlation coefficient has been computed between two adjacent pixels using the equation 6.
$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i$$ (3)
\[
D(x) = \frac{1}{N} \sum_{i=1}^{N} ((x_i - E(x))^2
\] (4)

\[
cov(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))
\] (5)

\[
\gamma_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}
\] (6)

Table [1] gives the correlation coefficient values of adjacent pixels in the Lena and Pepper images along horizontal, vertical and diagonal directions of the plain and encrypted images. This confirms that the adjacent pixels in the plain images are strongly correlated where as the adjacent pixels in the encrypted images are weakly correlated. Figure 7 illustrates the correlation distribution of the plain and encrypted Lena, Pepper as well as House images. Correlations along horizontal, vertical and diagonal directions are tested for Lena, Pepper and House images respectively. We can clearly see that the encrypted images are very weakly correlated.

| Images                  | Horizontal | Vertical | Diagonal |
|-------------------------|------------|----------|----------|
| Original Lena Image     | 0.9746     | 0.9381   | 0.9438   |
| Encrypted Lena Image    | -0.0184    | 0.0352   | 0.0228   |
| Original Pepper Image   | 0.9143     | 0.9403   | 0.9552   |
| Encrypted Pepper Image  | 0.0110     | 0.0065   | -0.0104  |

**Table 1:** Correlation coefficients between adjacent pixels for original and encrypted images.

5.4.2 Non-Parametric Test for Pixel Randomness

Evaluation results of the proposed non-parametric test for the house images are given in figure 8. Scores of the other tests images are given in the table 2. As one can see that the perfectly encrypted and perfectly shuffled images have high scores. In contrast the original image has very low score and the partially shuffled and encrypted images have average scores. Scores of the different images are varied according to their pixel randomness. Images with high scores are indistinguishable from random patterned images.

| Images       | Original | Partially Shuffled | Partially Encrypted | Perfectly Shuffled | Perfectly Encrypted |
|--------------|----------|--------------------|---------------------|-------------------|---------------------|
| Lena Image   | 1.3      | 65.4               | 65.8                | 97.6              | 98.8                |
| Boat Image   | 2.1      | 61.2               | 65.7                | 98.6              | 97.5                |
| Pepper Image | 2.6      | 66.3               | 67.4                | 97.4              | 98.5                |

**Table 2:** Scores of Test Images.
Figure 7: Correlation Distribution of the pairs of adjacent pixels: (a) Horizontal. (b) Vertical. (c) Diagonal

Figure 8: (a) Original (b) Partially Shuffled (c) Partially Encrypted (d) Perfectly Shuffled (e) Perfectly Encrypted

6 Conclusion

We have introduced a new test using non-parametric method in statistics for measuring the randomness in the image pixels. It scores the images, depending upon how far their pixels are i.i.d. The newly designed 2D CA based encryption scheme first analysed by standard evaluation methods and then the encrypted images are used to validate our newly proposed non-parametric test. Simulation results clearly show that this test can be used for evaluating the quality of any image shuffling and image encryption method. The complexity of the method is $O(NThn)$, so it can be efficiently implemented. In the proposed non-parametric test we have only used the basic run test to measure the randomness.
References

[1] BalaSuyambu Jeyaram, Rama Raghavan, Krishna Shankara Narayanan, New CA based Key Generation for a Robust RGB Color Image Encryption Scheme, International Journal of Computer Applications(0975 8887), Volume 80-No.7, October 2013, pp.45-50.

[2] BalaSuyambu Jeyaram, Rama Raghavan, New CA based image encryption-scaling scheme using wavelet transform, Journal of Systemsics, Cybermatics and Informatics, Volume 12- Number 3-Year 2014, ISSN: 1690-4524, pp.66-71.

[3] P. P. Choudhury, Birendra Kumar Nayak, Sudhakar Sahoo, Sunil Pankaj Rath, Theory and Applications of Two-dimensional, Null-boundary, Nine-Neighborhood, Cellular Automata Linear rules. CoRR abs/0804.2346 2008.

[4] P. Chattopadhyay, P. P. Choudhury, Characterisation of a Particular Hybrid Transformation of Two-Dimensional Cellular Automata, Computers and Mathematics with Applications, Elsevier publication, 38, 1999, pp.207-216.

[5] K. Dihidar, P. P. Choudhury, Matrix Algebraic formulae concerning some special rules of two-dimensional cellular automata, International Journal on Information Sciences, Elsevier publication, Volume 165, 2004, pp.91-101.

[6] J. D. Gibbons and S. Chakraborti, Nonparametric statistical inference, New York: Marcel Dekker, 1992.

[7] Li, X., A new measure of image scrambling degree based on grey level difference and information entropy., International Conference on Computational Intelligence and Security, IEEE,1, 2008,pp.350-354.

[8] N. H. Packard and S. Wolfram, Two-dimensional cellular automata, Journal of Statistical Physics, 38 (5/6), 1985, pp.901-946.

[9] W. Pries, A Thanailakis and H. C. Card, Group properties of cellular automata and VLSI Applications, IEEE Trans on computers C-35, December 1986, pp.1013-1024.

[10] Rama. R, Bala Suyambu. J, A. Arokiaraj, S. Saravanan, A study of DES Algorithm with Cellular Automata, International Journal of Innovative Management,Information and Production, Volume 3, Number 1, March 2012, ISME International 2011, ISSN 2185-5439, pp.10-16.

[11] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Leveson, M., Banks, D., Heckert, A., Dray, J., and Vo, S., A statistical test suit for random and pseudorandom number generators for cryptographic applications, NIST Special Publication, 800-22, 2010.

[12] J. von. Neumann, The Theory of Self-Reproducing Automata, (Edited by A. W. Burks) Univ. of Illinois Press Urbana, 1996.

[13] S. Wegenkittl, Entropy estimators and serial tests for ergotic chains., IEEE Trans. Inform. Theory, Vol. 47, no. 6, Sept. 2001, pp.2480-2489.

[14] S.Wolfram, Statistical mechanics of Cellular Automata, Rev Mod Phys. 55, July 1983, pp.601-644.
[15] Wu, Y., Zhou, Y., Saveriades, G., Agaian, S., Noonan, J., and Natarajan, P., Local shannon entropy measure with statistical tests for image randomness, Elsevier publication, Information Sciences 2012.

[16] Yue Wu, Sos Agaian and Joseph P. Noonan, ”A novel method of testing image randomness with applications to image shuffling and encryption” proc. SPIE 8755, Mobile Multimedia/Image Processing, Security and Applications, 2013.