Image Scrambling Methods For Digital Image Encryption

Raviraj Shelke
College of Engineering, Pune
Department of Electronics
Email: ravirajshelke49@gmail.com

Shilpa Metkar
College of Engineering, Pune
Department of Electronics
Email:metkarshilpa@gmail.com

***Abstract*—Information security is main concern at present. Image scrambling is one of the way to encrypt the digital image data. This paper present novel image scrambling method using Non-Commutative wavelet transform and Poker Shuffle transform. Due to suitability of Non-commutative Wavelet transform for image scrambling application over conventional Wavelet transform and non-linearity and non-analytic computation characteristics of Poker Shuffling increases the performance of proposed methodology. The proposed method compared with different existing image scrambling algorithm on the ground of image encryption.**

Keywords: Image scrambling, Non-commutative Wavelet Transform, Poker Shuffle Transform.

I. Introduction

Rapid development of multimedia technology, internet and digitization Video, Image and Audio data have been an important way to transmit information; hence encryption of these information sources is topic of great interest to researchers. Image scrambling is one of the way to encrypt the image data. Aim of image scrambling is to destroy the original image contents to make it difficult for the intruders to get original information out of it. So image scrambling transform the original image to random pattern image which is meaningless and imperceptible by human eyes. Recent years, many Image scrambling methods have been proposed. There are different approaches of image scrambling which will be discussed later in this section. The evaluation of scrambling degree states the security level of the algorithm. Greater the scrambling degree higher the security of encrypted image. Subjective evaluation of scrambling (i.e. using Human Visual System (HVS)) is uncertain[1]. Objective evaluation of an scrambling algorithm can be done through unchanged pixel positions, entropy, correlation etc.There are 2 main approaches of image scrambling:

I. Pixel position based scrambling

II. Pixel value based Scrambling.

In first approach, depending on the scrambling key, pixel positions of the original image are mapped to scrambled image. In this approach, as pixel values are not touched pixel distribution remains same as the original image. In second approach, pixel values are changed according to the scrambling key to get scrambled image. As this approach scramble the image data by transforming pixel values, pixel distribution of scrambled image is not same as original image. Therefore by objective evaluation, second approach is more secure as compared to first but at the cost of chance of losing image data. To get highly secure image scrambling algorithm one can effectively combine both approaches. There are also the scrambling methods available which scrambles the image by transforming original image to transform domain and then apply image scrambling algorithm to get scrambled image[2].

In this paper new scrambling algorithm is introduced which uses Non-commutative wavelet transform and Poker shuffling algorithm. For image scrambling, low frequency(LF) coefficients of Non-commutative wavelet are scrambled using Poker shuffling algorithm and high frequency(HF) coefficients kept unchanged. To get final scrambled image, Inverse Non-commutative wavelet transform is applied on the scrambled-LF and HF coefficients.

The organisation of paper is as follows. In section II, different scrambling algorithms are discussed. In section III, a brief introduction to the non-commutative wavelet transform is given. The main idea for image scrambling using Non-commutative wavelet transform and Poker Shuffling is discussed in section IV. In section V experimental results are given. Finally, section VI concludes the paper.

II. Scrambling Algorithms

A. Arnold Transform

Arnold transform is given by[3]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a00 & a01 \\ a10 & a11 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} ModN \tag{1}$$

where, $xandy$ represent the pixel position in original image and $x'andy'$ is the pixel position in new scrambling image and N is the size of image. To make image scrambling reversible, values of $a00, a01, a10anda11$ should satisfy the condition in equation2.

$$a00 * a11 + a01 * a10 = \pm 1 \tag{2}$$

B. Fibonacci Transform

Fibonacci transform is same as Arnold transform, only difference is that the values of $a00, a01, a10anda11$ are the consecutive 4 numbers in Fibonacci series[4]. Fibonacci series is given as follows:

$$F = \begin{cases} 0 & \text{if } n < 1; \\ 1 & \text{if } n = 1; \\ F(n-1) + Fn - 2 & \text{if } n > 1 \end{cases} \quad (3)$$

C. Queue Transform

Image of size $M \times N$ contains M rows and N columns. If we consider row matrix, then image is having M queues of length N. Similarly if we consider column matrix then it has N queues of length M. Algorithm for Queue transform is discussed as follows. Let us consider reference point for scrambling (R, S) which is also acts as key for image scrambling. The algorithm for Queue transform[5], [6] is as follows:

i) Consider row matrix as queue. If row number (i) is greater than (R) then move left $(i - R)$ column. If row number(i) is less than (R) then move right $(R - i)$ column.
ii) Consider column matrix as queue. If column number (j) is greater than (S) move up $(j - S)$ row. If column number(j) is less than (S) move down $(S - j)$ row.
iii) Repeat (ii) and (iii) for given number of iterations to get scrambled image.

Step(ii) and step(iii) is individually called One Queue transform. After each One Queue transform, transformed queue contains every element from different queue. We have demonstrated Queue Transform on matrix U of size (5×5) and using reference point $(R, S) = (3, 4)$ which is given in equation4.

$$U = \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \\ 21 & 22 & 23 & 24 & 25 \\ 31 & 32 & 33 & 34 & 35 \\ 41 & 42 & 43 & 44 & 45 \\ 51 & 52 & 53 & 54 & 55 \end{bmatrix} \quad (4)$$

After application of one Queue Transform on each row of matrix U, as discussed in step(i), we get matrix V given in equation5.

$$V = \begin{bmatrix} 14 & 15 & 11 & 12 & 13 \\ 25 & 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 & 35 \\ 42 & 43 & 44 & 45 & 41 \\ 53 & 54 & 55 & 51 & 52 \end{bmatrix} \quad (5)$$

After application of one Queue Transform on each column of matrix V, as discussed in step(ii), we get matrix Z given in equation6

$$Z = \begin{bmatrix} 31 & 43 & 55 & 12 & 24 \\ 42 & 54 & 11 & 23 & 35 \\ 53 & 15 & 22 & 34 & 41 \\ 14 & 21 & 33 & 45 & 52 \\ 25 & 32 & 44 & 51 & 13 \end{bmatrix} \quad (6)$$

We can observed from equation6 that each element in Matrix Z is from different row and column. To get scrambled image same steps are followed.

D. Image Scrambling Using Logistic Map Chaotic Sequence

Chaos theory is field of study in mathematics that studies the dynamic systems behaviour and condition that are highly sensitive to initial condition. Slight change in initial condition leads to widely diverging output. In other words if chaos system is used for image scrambling then without knowledge of initial condition (here it is scrambling key) one may not be able to descramble the image from input scrambled image.

There are many chaotic maps which are used for generating chaotic sequence. one of them is Logistic map. Logistic map is a polynomial mapping in which chaotic behaviour is generated by nonlinear dynamic equation given in equation7. Logistic map is given by:

$$x_(n+1) = \mu x_n(1 - x_n) \quad (7)$$

where, $0 < x_n < 1$ where, $n = \{0, 1, , N\}$, with $x_0 = initialcondition$; and $\mu = Bifurcationparameter$.

To get chaotic behaviour from Logistic map[7] value of μ must be in range $3.5699456 < \mu < 4$. Algorithm for image scrambling using Logistic map is as follows:
Consider an image I of size is $M \times N$ and (i, j) represent the pixel coordinates, $i = (1, 2, , M)$ and $j = (1, 2, , N)$

i) Generate chaotic sequence, Sq, of length $M \times N$ using equation7. Values of x_0 ,μ can be used as scrambling key.
ii) Scale the chaotic sequence values from range $0 < Sq < 1$ to the range $0 < Sq' < 255$.
iii) Create 2D array $R(i, j)$ of size $M \times N$ from 1D sequence Sq'.
iv) Perform EX-OR operation which is reversible operation; on pixel values and array $R(i, j)$ to get pixel value transformed image I'.

$$I'(i, j) = I(i, j) \oplus R(i, j) \quad (8)$$

v) Apply one of the scrambling algorithm discussed in section II-A, section II-B, section II-C to get final scrambled image.

To descramble the image, $key = \{x_0, \mu\}$ is needed. This method is comparatively more secure as compared other methods discussed in section II-A, section II-B and section II-C[8] because it combines advantages of both, pixel position based scrambling and pixel value based scrambling methods.

E. Poker Shuffling Transform

Poker Shuffle transformation[9] is basically pixel position shuffling technique in which shuffling process is controlled by chaotic sequence map. Idea of Poker shuffle comes from Poker game. Card shuffling in poker game plays important role in randomizing the cards and to make it unpredictable to guaranteed the fairness of game. Similar approach is used for pixel position shuffling. Poker Shuffling transformation consist of 3 operations[9]:

1) Card Inter-Crossing Operation $S_sq(P_1, Q_1)$: In card inter-crossing operation given sequence Sq is divided into 2 sub-sequences $Sq1$ and $Sq2$. These 2 sequence are merged into one sequence according to $key = \{P_1, Q_1\}$.

For example, if Sq is given as:

$$Sq = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$$

then $Sq1$ and $Sq2$ will be,

$$Sq1 = [1, 2, 3, 4, 5, 6]$$

$$Sq2 = [7, 8, 9, 10, 11, 12, 13]$$

If $key = \{P_1, Q_1\} = \{3, 2\}$ then output of Card Inter-Crossing is:

$$S_sq(2, 1) = [7, 8, 1, 9, 10, 2, 11, 12, 3, 13, 4, 5, 6]$$

Similarly,

$$S_sq(3, 2) = [7, 8, 9, 1, 2, 10, 11, 12, 3, 4, 13, 5, 6]$$

2) Card Extraction Operation $D_sq(P_2, Q_2)$: For card extraction process make sure that the value of P_2 should be less than Q_2, otherwise exchange the values of P_2 and Q_2. In this process, from original sequence:

$$Sq = (1, 2, 3...., P_2-1, P_2, P_2+1, ..., Q_2-1, Q_2, Q_2+1, ...)$$

subsequence $(P_2, ..., Q_2)$ is extracted and appended at the start of the sequence.

For example,if Sq is given as:

$$Sq = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$$

then,

$$D_sq(P_2, Q_2) = (6, 10) = [6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 11, 12, 13]$$

3) Card Cutting Operation $T(P_3)$: This operation is nothing but Card extraction operation $D_sq(P_2, N)$, where N is the last element in the sequence.

For example,if Sq is given as:

$$Sq = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$$

in which $N = 13$ then,

$$T(8) = D_sq(8, 13) = [8, 9, 10, 11, 12, 13, 1, 2, 3, 4, 5, 6, 7]$$

These 3 operations are used to scramble image in Poker Shuffling Transformation. Algorithm for Poker Shuffling transform is as follows. Let us first conside input image I is of size $M \times N$:

i) Construct the keys $P_{(1...n)}$ and $Q_{(1...n)}$ for Poker shuffle operation using any of the chaotic map.
ii) Construct 1D array R of sequence $R = (1, ..., M \times N)$
iii) Using Poker shuffle operations (discussed in section II-E1, section II-E2 and section II-E3) and scrambling keys, scramble input array R to get scrambled array, R'.
iv) Convert input image I to 1D array, I', and map the pixel index in I' according to R' to get scrambled pixel array $I1'$. Convert $I1'$ to 2D pixel array to get scrambled image.

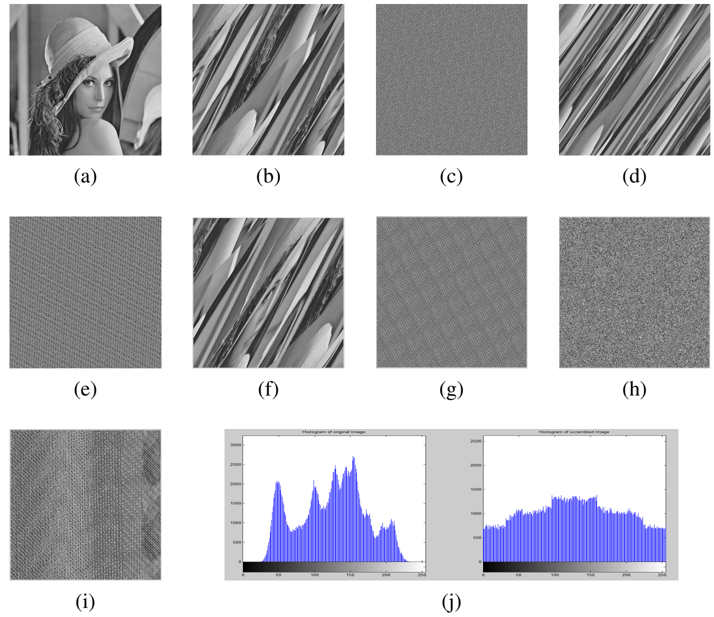


Figure 1: (a)Original Image(512×512), (b)Arnold Transform(iteration=1), (c)Arnold Transform(iteration=10), (d)Fibonacci Transform(iteration=1), (e)Fibonacci Transform(iteration=10), (f)Queue Transform(iteration=1), (g)Queue Transform(iteration=10), (h)Logistic Map Chaotic Sequence, (i)Poker Shuffling, (j)Pixel Value Distribution in Original image(left side) and Scrambled Image Using Logistic Map Based Chaotic Sequence(right side).

Analysis of all algorithms discussed in section II are given in section V. Out of these 5 algorithms (section [II-A – II-E]), only algorithm discussed in section II-D is does both, pixel position shuffling and pixel distribution transformation. Algorithm mentioned in section II-A and section II-B are periodic[10]; means after certain iteration we get same image as scrambled image, so they are less secure as compared other algorithms. Poker shuffling in section II-E is more secure as it very complex to decode because it combines the advantages of chaotic map and statistical independency in shuffling process.

III. Non-Commutative Wavelet Transform

In conventional wavelet transform, in each level of decomposition image is divided into 4 sub-images, which are called as approximation, vertical, horizontal and diagonal wavelet sub-bands respectively. Out of these 4 sub-bands approximation sub-band contains LF wavelet coefficients and other three are HF sub-bands. HVS is less sensitive to diagonal details and most sensitive to LF details. As mentioned in [11] using non-commutative wavelet analysis we can decompose the image like conventional wavelet transform in 4 sub-bands but 2 LF sub-bands and 2 HF sub-bands by discarding diagonal details in conventional wavelet transform. For Non-commutative wavelet decomposition and reconstruction algorithm given in [12] is used, which is as follows. Consider image I of size $M \times N$, and after decomposition we get 4 sub-bands of wavelet coefficients namely $L1(i,j), L2(i,j), H1(i,j) and H2(i,j)$; where $1 < i < \frac{M}{2}$, $1 < j < \frac{N}{2}$. The Non-commutative wavelet transformation and reconstruction equations are given below: Decomposition Equations:

$$L1(i,j) = \triangle \times [I(2i-1, 2j-1) + I(2i, 2j)] \tag{9a}$$

$$H1(i,j) = \triangle \times [I(2i-1, 2j-1) - I(2i, 2j)] \tag{9b}$$

$$L2(i,j) = \triangle \times [I(2i-1, 2j) + I(2i, 2j-1)] \tag{9c}$$

$$H2(i,j) = \triangle \times [I(2i-1, 2j) - I(2i, 2j-1)] \tag{9d}$$

Reconstruction Equations:

$$I(2i-1, 2j-1) = \Psi \times [L1(i,j) + H1(i,j)] \tag{10a}$$

$$I(2i-1, 2j) = \Psi \times [c(i,j) + H2(i,j)] \tag{10b}$$

$$I(2i, 2j-1) = \Psi \times [c(i,j) - H2(i,j)] \tag{10c}$$

$$I(2i, 2j) = \Psi \times [L1(i,j) - H1(i,j)] \tag{10d}$$

Reconstruction of original image without error, is dependent on the values of $\triangle$ and Ψ. As mentioned in[12], when $\triangle = 1$, $\Psi = \frac{1}{2}$ or $\triangle = \frac{1}{2}$, $\Psi = 1$ image I can be reconstructed without error.

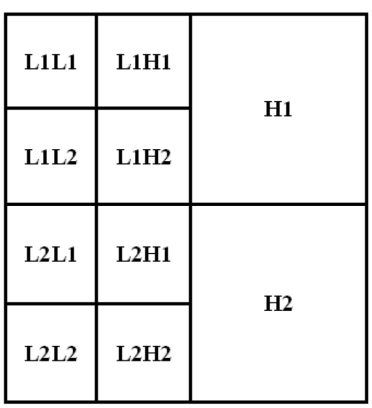
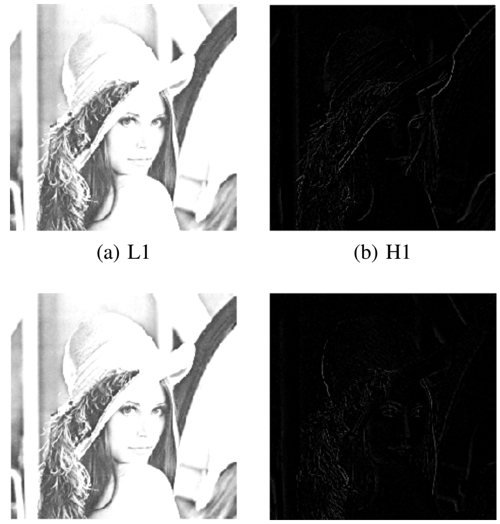


Figure 2: Two-Level Non-Commutative Wavelet Transform Decomposition

In Figure 2, $L1$ and $L2$ represents LF sub-bands and $H1$ and $H2$ represent HF sub-bands. To decompose further, in conventional wavelet transform approximate(LF) sub-band is used to decompose the image to further levels. In non-commutative wavelet transform we can also use L2 sub-band to decompose image to further level. In this paper only single level decomposition is used.



(a) L1 (b) H1

(c) L2 (d) H2

Figure 3: Wavelet Sub-band After First Level of Decomposition

IV. Proposed Methodology

Poker shuffling algorithm from section-II-E is combined with Non-Commutative wavelet transform discussed in section-III for the formation of proposed methodology. As most of the perceptible information contained in LF coefficients, in proposed algorithm LF sub-bands of Non-Commutative wavelet transforms are scrambled using Poker shuffling algorithm and then inverse Non-commutative wavelet transform is applied to the sub-bands to get scrambled image. As this operation is performed in wavelet domain and scrambled image is in spatial domain it becomes very hard for the intruders to decode the image and therefore we get highest level of security. Algorithm for proposed methodology is as follows.

i) Apply 1-level Non-Commutative wavelet transform on input image to get 4 sub-images $L1, L2, H1$ and $H2$ of size $\frac{M}{2} \times \frac{N}{2}$, where $MandN$ represents the height and width of the original input image respectively.
ii) Generate scrambling pattern R using Poker shuffling algorithm of length $\frac{M}{2} \times \frac{N}{2}$ given in section-II-E and shuffle $L1$ and $L2$ using Poker shuffling algorithm to get $L1'$ and $L2'$.
iii) Apply inverse Nom-commutative wavelet transform on sub-bands $L1', L2', H1$ and $H2$ to get scrambled image.

As LF sub-bands of Non-commutative wavelet Transform are scrambled, the perceptual information is effectively destroyed from original image. To recover the original image from scrambled image exctly reverse steps are followed. The experimental results of Non-commutative wavelet transformation and scrambling using Non-commutative wavelet transform are discussed in next section.

TABLE I: Comparison of Scrambling Algorithm w. r. t. Time(sec)

Name	Arnold	Fibonacci	Poker Shuffle	Queue Trans.	Logistic Map
Lena.bmp (512 × 512)	2.6047	2.2097	0.79689	0.5879	0.3713
Bmw.jpeg (320 × 320)	1.6194	1.1075	0.8081	0.7740	0.2942
Test1.jpg (3264×2448)	18.4661	18.5876	19.9358	16.6707	12.3263
Test2.jpg (2816×2112)	13.8200	13.8151	13.9998	11.0223	9.3511
Test3.jpg (2816×2112)	13.8259	13.6839	14.3953	10.0185	9.2676
Test4.jpg (1536×2048)	7.2837	7.1847	9.5666	13.0185	5.2694
Test5.jpg (2816×2112)	14.8931	13.8339	14.4472	15.0156	9.3025

V. Experimental Results

All the scrambling methods mentioned in section II are implemented and compared with respect to required timing for the scrambling process. TABLE I gives the timing requirement(in sec.) for each algorithm. Also the results of scrambling on input image are given in Figure 1

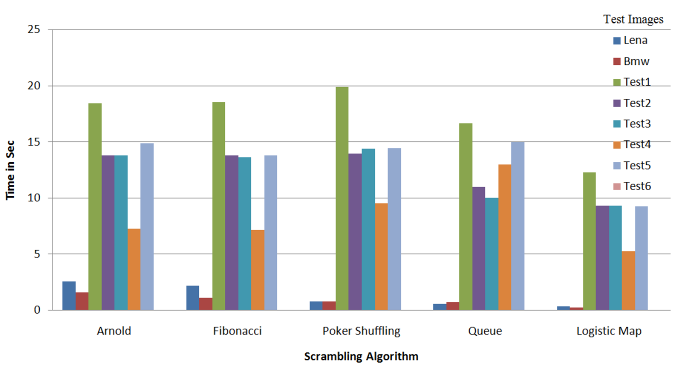


Figure 4: Plot of Scrambling Methodology Vs Timing Requirement

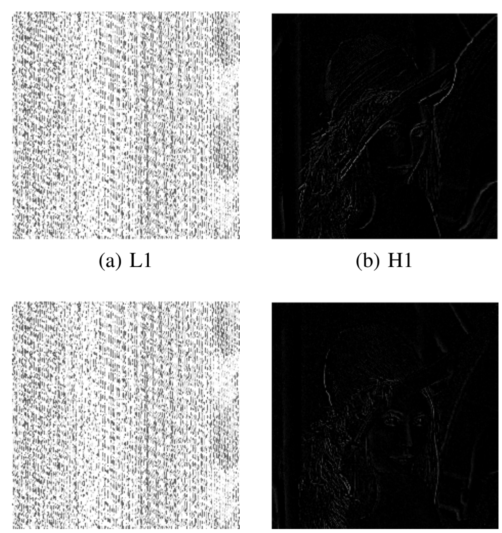
From TABLE-I and Figure 4 it can be observed that Arnold and Fibonacci transformation required same amount of time for image scrambling. Also these 2 scrambling algorithms along with Queue transform are periodic[10] means at specific number of iteration we get same input image as scrambled image.

Logistic map require least time compared to other methods. But because it modifies the pixel value for getting scrambled image we can't use this method with Non-commutative wavelet in proposed methodology, as it will make wavelet transformation irreversible.

Poker Shuffling Transformation required maximum amount of time as compared to other methods, but as it will modifies pixel positions for getting scrambled image unlike Logistic map and due to non-periodic, non-algebric properties of Poker shuffling transform is used in proposed methodology.

TABLE II: Comparison of Scrambling Methods Using Non-Commutative Wavelet Transform + Different Scrambling Algorithm w. r. t. Time(sec)

Name	Non Commu. wavelet+ Arnold	Non Commu. wavelet+ Fibonacci	Non Commu. wavelet+ Queue Trans.	Proposed Method
Lena.bmp (512 × 512)	1.0083	1.0009	0.8959	1.0311
Bmw.jpeg (320 × 320)	1.2000	1.8939	1.0390	1.9889
Test1.jpg (3264×2448)	38.5939	36.8510	33.1611	40.9144
Test2.jpg (2816×2112)	28.1757	28.3890	24.1253	33.3384
Test3.jpg (2816×2112)	28.1686	27.7394	23.3421	34.0033
Test4.jpg (1536×2048)	15.8234	27.4201	23.3324	34.3067
Test5.jpg (2816×2112)	28.3747	27.4201	23.3324	34.3067



(a) L1 (b) H1

(c) L2 (d) H2

Figure 5: Image After Scrambling Low Frequency Coefficients L1 and L2 in Figure 3.

Image scrambling using Non-commutative wavelet transform is achieved by scrambling the LF sub-bands of Non-commutative wavelet using one of the method discussed in section II. TABLE-II gives the comparison of these scrambling methods when used for scrambling the LF sub-bands of Non-Commutative wavelet transform.

As scrambling using Logistic Map changes the pixel distribution of original image, this method can not be combined with Non-commutative wavelet transform for image scrambling. From TABLE-II, we can observed that pro-

TABLE III: Comparison of Proposed Method with normal Poker Shuffling Transform w. r. t. SSIM

Name	Proposed Method	Poker Shuffle Transform
Lena.bmp (512×512)	0.0248	0.0607
Bmw.jpeg (320×320)	0.0244	0.0404
Test1.jpg (3264×2448)	0.0794	0.177
Test2.jpg (2816×2112)	0.0254	0.0534
Test3.jpg (2816×2112)	0.0288	0.0154
Test4.jpg (1536×2048)	0.044	0.349
Test5.jpg (2816×2112)	0.0257	0.0485

posed method(Non-commutative wavelet+Poker Shuffle) requires maximum time compared to all other scrambling techniques, but due to the advantages of poker shuffling over other scrambling techniques it is selected in proposed method.

Figure 5 shows the wavelet sub-bands after scrambling LF coefficients of Non-commutative wavelet shown in Figure 3a and Figure 3b using Poker shuffling. Figure 6 shows the scrambled image in spatial domain after applying inverse wavelet transform.

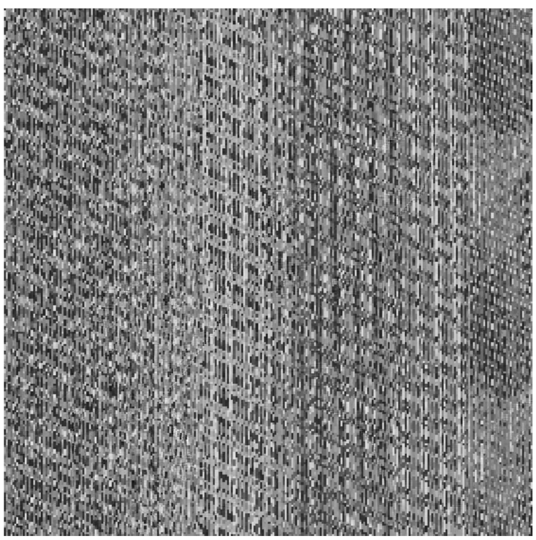


Figure 6: Scrambled Image After Application Inverse Non-Commutative Wavelet Transform

TABLE-III gives comparison of proposed method with normal poker shuffling transform with respect to Structural Similarity(SSIM)[13]. SSIM indicates the performance of proposed methodology with respect to security. Less SSIM indicates higher security and vice-versa.

VI. CONCLUSION

This paper proposed new scrambling method using Non-commutative Wavelet Transform and Poker Shuffle Transformation. As image scrambling is done in wavelet space and due to non-algebraic, non-periodic Poker Shuffle Transformation we get highly secure image scrambling algorithm compared to other scrambling methods. Though timing requirement for Poker Shuffle Transformation is more as compared to other scrambling methods, by using same scrambling sequence for both LF component and reducing no. of iterations we can reduce overall timing requirement of proposed method.

REFERENCES

[1] X. Li, *A New Measure of Image Scrambling Degree Based on Grey Level Difference and Information Entropy*, International Conference on Computational Intelligence and Security, 2008, pp. 350-354.

[2] Y. Chen and W. Zhang, *A New Image Scrambling Technology Based On Noncommutative Wavelets Domain*, International Symposium on Computer Science and Society, 2011, pp. 20-22.

[3] R. Ye, *A Novel Image Scrambling and Watermarking Scheme Based on Orbits Of Arnold Transform*, Pacific-Asia Conference on Circuits, Communication and System, 2009, pp. 485-488.

[4] W. Zou, J. Huang and C. Zhou, *Digital Image Scrambling Technology Based On Two Dimension Fibonacci Transformation And Its Periodicity*, Third International Symposium on Information Science and Engineering, 2010, pp. 415-418.

[5] H. Zhang, *A New Image Scrambling Algorithm Based On Queue Transformation*, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, August 2007, pp. 1526-1520, .

[6] H. Zhang, *A New Image Scrambling Algorithm*, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, July 2008, pp. 1088-1092.

[7] P. Grassberger and I. Procaccia, *Measuring the Strangeness of Strange Attractors*, Physics D9. May 1983, pp. 189-208.

[8] W. Yanling, *Image Scrambling Method Based On Chaotic Sequences and Mapping*, First International Workshop on Education Technology and Computer Science, 2009, pp. 453-457.

[9] X. Wang and J. Zhang *An Image Scrambling Encryption Using Chaos-Controlled Poker Shuffle Operation*, IEEE 2008.

[10] Z. Shang, H. Ren and J. Zhang, *A Block Location Scrambling Algorithm of Digital Image Based on Arnold Transformation*, The 9th International Conference for Young Computer Scientists. 2008, pp. 2942-2947.

[11] W. Guo, *Unitary Bases and Non-Commutative Wavelets*, Acta Mathematica Sinica, English Series, vol. 19(3), pp. 591-594, July 2003.

[12] Y. Chen and W. Zhang, *A New Image Scrambling Technology Based On Noncommutative Wavelets Domain*, International Symposium on Computer Science and Society, 2011, pp. 20-22.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Trans. Image Processing, vol. 13, no. 4, pp. 600-612, 2004.