2012 International Conference on Information and Computer Networks (ICICN 2012)
IPCSIT vol. 27 (2012) © (2011) IACSIT Press, Singapore

A new method of image encryption with multiple-parameter discrete fractional Fourier transform

Mohammad Monajem [1], Shahriar Baradaran Shokouhi [2]

[1] Iran University of Science and Technology

Abstract. In this paper a new method for image encryption has been proposed. In this method multi-parameter discrete fractional Fourier transform (MPDFRFT) has been used. Our method has been compared with the proposed method by Pei in 2009 that is one of the recent methods in this branch. Simulation results show image encryption criteria are provided. Also speed of algorithm, percentage of fixed points and correlation coefficient between original image and decrypted image has been improved in comparison with the method of Pei.

Keywords: Commuting matrix, discrete Fourier transform, discrete fractional Fourier transform, eigenvector.

1. Introduction

The discrete fractional Fourier transform (DFRFT) is a generalization of the DFT with additional free parameters [1] – [3]. In [1], a DFRFT with one fractional parameter was defined by taking fractional eigenvalue powers of an eigendecomposition of the DFT matrix. The DFT eigenvectors used in [1] are Hermit –Gaussian like. These eigenvectors are computed from a DFT –commuting matrix proposed in [4]. As a result, the DFRFT can approximate samples of continuous fractional Fourier transform (FRFT) [5]. Continues FRFT is useful for optics and signal processing [5] , [6]. In [3], Pei and Hsue extended the work in [1] by taking different fractional powers for different eigenvalues of the DFT and define a new multiple-parameter DFRFT (MPDFRFT) with N fractional order parameters, where N is the number of input data points. On the other hand, the DFT is randomized to define the discrete random Fourier transform (DRFT) in [7] by taking random powers for eigenvalues of the DFT matrix. The eigenvectors of DRFT are not random because they are Hermit- Gaussian like and are computed using the same method proposed in [1].

This paper is organized in five sections. In section 1 the introduction was mentioned. In section 2, preliminaries like DFT matrix, EDFT, RDFRFT and MPDFRFT formulas are defined. Section 3 explains the proposed method of this paper. In section 4, simulations and results of this method are discussed. Finally in 5 conclusions of paper are mentioned.

2. Preliminaries

Definition: Let T be N×N transform matrix and $\mathbf{x}$ be an N×1 input data vector. Assume that its transform output y=Tx is an N × 1 vector with complex entries. If $y=[y_0, y_1, \ldots, y_{N-1}]^T$, then $[|\mathbf{y_0}|, |\mathbf{y_1}|, \ldots, |\mathbf{y_{N-1}}|^T]$ is called the output magnitude and $[\angle y_0, \angle y_1, \ldots, \angle y_{N-1}]$ is called the output phase, where T denote the transpose operation and $\angle y_i$ is the phase of y_i.

The N×N DFT matrix F is defined as:

$$[F]_{m,n} = \frac{\mathbf{1}}{\boldsymbol{N}} \boldsymbol{e^{-j\,(2\pi/N)mn}}, \qquad 0 \le m,n \le N-1 \tag{1}$$

Eigendecomposition of F is defined as blow:

[1] Mohammad Monajem. Tel.: + 989163054419
E-mail address: monajem.mohamad@gmail.com

$$F= \sum_{k=0}^{N-1} \boldsymbol{\lambda_k}\ \mathbf{e_k}\ \mathbf{e_k^T} \tag{2}$$

Where $e_0, e_1, \ldots, e_{N-1}$ are orthogonal eigenvectors of DFT. F has only four eigenvalues that contain {1, -j , -1, j} [8] . The DFRFT F^a with one order parameter "a" is defined by [1]:

$$\boldsymbol{F^a} = \sum_{k=0}^{N-1} \boldsymbol{\lambda_k^a}\ \mathbf{e_k}\ \mathbf{e_k^T} \tag{3}$$

From [3], the MPDFRFT, which is a generalization of the DFRFT, can be defined as:

$$\boldsymbol{F^{\bar{a}}} = \sum_{k=0}^{N-1} \boldsymbol{\lambda_k^{a_k}}\ \mathbf{e_k}\ \mathbf{e_k^T} \tag{4}$$

Where the 1×N parameter vector $\bar{\mathbf{a}}$ is defined as $\bar{\mathbf{a}} = [a_0, a_1, \ldots, a_{N-1}]$. In [3] random eigenvalues of MPDFRFT, $\boldsymbol{\lambda_k^{a_k}}$ are defined as below:

$$\lambda_k^{a_k} = \begin{cases} \left(e^{-j2\pi}\right)^{ak} & \text{if } \lambda_k = 1 \\ \left(e^{-\frac{j\pi}{2}}\right)^{ak} & \text{if } \lambda_k = -j \\ \left(e^{-j\pi}\right)^{ak} & \text{if } \lambda_k = -1 \\ \left(e^{-j3\pi/2}\right)^{ak} & \text{if } \lambda_k = j \end{cases} \tag{5}$$

We defined $\boldsymbol{\lambda_k^{a_k}}$ in a deifferent way presented in the next parts. MPDFRFT has random eigenvectors. It has random magnitude and random phase. It is a suitable property to provide image encryption criteria.

3. Proposed DFRFT with random eigenvectors and eigenvalues

In this section definition of DFT-commuting matrix with orthonormal random DFT eigenvectors should be discussed. Before this definition, we define commuting matrix. Assume A and B are arbitrary matrices, and AB=BA. In This status, we say A is B- commuting matrix (or A commutes with B). A and B have same eigenvectors but different eigenvalues.

Definition: An N×N matrix A is K-symmetric [9] if KAK=A, where K is a reversal matrix with below definition:

$$K=F^2=F^{-2} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J_{N-1}} \end{bmatrix} \tag{6}$$

$\mathbf{J_{N-1}}$ is a (N-1) × (N-1) reversal matrix whose only nonzero entries are ones on the antidigonal.

Theorem: If an N×N matrix C is defined as:

$$C=M+FMF^{-1} \tag{7}$$

Where M is an arbitrary N×N K-Symmetric matrix, then C commutes with F [10].

in [11] a method for creating F-commuting matrix is proposed. This method is as below:

1) Form an N×N real random matrix D, whose N×N entries are independent and uniformly distributed over the interval [-1,1]. 2) Take the symmetric part of E to form the random matrix D, i.e.: E= (D+KDK)/2. 3) Take the symmetric part of E to form the random generating matrix G:

$$G= \frac{\mathbf{E+\ E^T}}{\mathbf{2}} \tag{8}$$

Where T denotes the matrix transpose operation. 4) Substitute M in (7) by G in (8) to construct the random DFT- commuting matrix H:

$$H= G+ FGF^{-1} \tag{9}$$

We know H commutes with F. in [11] MPDFRFT is constructed using H as below:

$$\boldsymbol{F_H^{\bar{a}}} = \sum_{k=0}^{N-1} \boldsymbol{\lambda_k^{a_k}}\ \mathbf{r_k}\ \mathbf{r_k^T} \tag{10}$$

Where r_k are orthonormal random DFT eigenvectors computed from H and $\boldsymbol{\lambda_k^{a_k}}$ is given by (5).

As it said in definition of commuting matrix, if A commutes with B then they have same eigenvectors but different eigenvalues. H commutes with F. So its eigenvalues are different with F. Also F has only four eigenvalues that are {1, -j, -1, j}. So DFT eigenvalues have not any variety. But it is not true about H. H eigenvalues are various and we can use this property to provide more randomized eigenvalues for MPDFRFT ($\mathbf{F_H^{\bar{a}}}$) . So we difine $\boldsymbol{\lambda_k^{a_k}}$ with a different way as below:

$$\boldsymbol{\lambda_k^{a_k}} = (\boldsymbol{\lambda_k})^{\mathbf{ak}} \tag{11}$$

This definition for $\boldsymbol{\lambda_k^{a_k}}$ is simpler than (5) and provides randomization property better. So we use (10) that $\boldsymbol{\lambda_k^{a_k}}$ is defined on it as (11). In proposed method we used 2-dimentional MPDFRFT. It means we used MPDFRFT twice to encrypt image. If P be the original image, then 2-D encryption is as below:

$$\mathbf{P}_{(\bar{a}_1,H1,\bar{a}_2,H2)} = \mathbf{F_{H_1}^{\bar{a}_1}} . \mathbf{P} . \mathbf{F_{H_2}^{\bar{a}_2}} \tag{12}$$

Where $\mathbf{F_{H_1}^{\bar{a}_1}}$ and $\mathbf{F_{H_2}^{\bar{a}_2}}$ are the N×N and M×M MPDFRFT matrices respectively. Also P is N×M image. For decryption, $(\mathbf{F_{H_1}^{a_1}})^{-1}$ and$(\mathbf{F_{H_2}^{\bar{a}_2}})^{-1}$ are used as below:

$$Q = (\mathbf{F_{H_1}^{\bar{a}_1}})^{-1} . \mathbf{P}_{(\bar{a}_1,H1,\bar{a}_2,H2)} . (\mathbf{F_{H_2}^{\bar{a}_2}})^{-1} \tag{13}$$

In the next part we discuss about simulation result of this method.

4. Simulations and Results

In this section simulation results are discussed and it will be seen that image encryption criteria are provided and some of them improved by proposed method.

4.1. Property of information hiding

One of the most important properties that should be provided by an affective image encryption algorithm is information hiding; as no information of original image can be gotten from the encrypted image. We encrypted a standard image by our method and its result is as below.

a b

Fig.1 (a) original image (cameraman (256*256), (b) encrypted image by proposed method.

As we see in Fig.1 our method encrypt image successfully and no information of original image can be gotten. So this method provides this property very well.

4.2. Speed of algorithms

A good image encryption algorithm should be fast and does encryption in a short time; although this time is different for different methods. We can compare our method with last method in this branch and if it was less than it, we can say our method is fast. We compared this time with the algorithm of [11] that proposed in 2009. Below table shows the program running time for this method.

Table 1. Program running time

Algorithm	Running time

Method of [11]	3.3455 sec
Proposed method	2.8986 sec

Table 1. shows our method is faster than method of [11]. It is because of changing that we made in $\lambda_k^{a_k}$ definition and defined it very sampler than [11]. In [11] $\lambda_k^{a_k}$ was defined by (5) and we defined it $(\lambda_k)^{a_k}$. This definition decreases complexity of program and causes less time for running.

4.3. Percentage of fixed points

One of the criteria for an image encryption algorithm is the percentage of fixed points (PFF). Assume P(i,j) is the original image with length of H and width of W. Also E(i,j) is the encrypted image of P. we know P and E have the same size. We define D(i,j) as below:

$$D(i,j) = \begin{cases} 1 & |P(i,j) - E(i,j)| < \delta \\ 0 & \text{others} \end{cases} \quad (14)$$

With respect to (14) the corresponded points in P and E that their difference is less thanδ, are known as "Fixed points". PFP is calculated by the following formula:

$$PFP = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} D(i,j)}{W \times H} \times 100 \quad (15)$$

It is clear that smaller value for PFP is better than bigger one and shows low similarity between original image and decrypted image. We calculated PFP for method of [11] and our proposed method. The results are as below table. δ is equals to 10 in this calculations.

Table 2. PFP for proposed method and method of [11] by δ =10 for "cameraman" image (256*256)

algorithm	PFP value
Method of [11]	6.73%
Proposed method	0.6027%

According to this table PFP of our proposed method is ten times smaller than method of [11]. It is because of changing that we made in definition of $\lambda_k^{a_k}$. As we explained before, with this definition, randomization property is provided better than method of [11] and this gets better results for image encryption.

4.4. Key space

In our proposed method key space is large enough and is N6×1016 for an N×N image. So our method is resistant against brute force attack. Brute force attack is an attack that unauthorized person tests all possible keys to find the encryption key. When the key space is large enough, brute force attack will be unuseful for unauthorized person.

4.5. Correlation Analysis

Image has special properties like that its neighbor pixels are correlated to each other highly. A good image encryption algorithm must decrease this correlation. In table 3 you can see correlation coefficient between one pixel of image with all pixels of it. This coefficient is calculated by below formula:

$$\text{Correlation coefficient} = \sum \sum \frac{(x_i - \mu x)(y_i - \mu y)}{\delta_x \delta_y} \quad (16)$$

In this formula x_i and y_i are intensity values of gray scale for two neighbor pixels, μ_x is the mean for x_i values and δ_x is standard deviation for x. μ_x and δ_x are calculated by below formulas:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (17)$$

$$\delta_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)^2} \quad (18)$$

With these formulas below values are gotten for correlation coefficients.

Table 3. Correlation coefficient between one pixel and whole of image.

algorithm	correlation coefficient
Original image	0.0654
Encrypted image by proposed method	-2.34×10^{-4}

As this table shows our proposed method decreases correlation coefficient very well and makes it negative that means correlation between pixels in encrypted image is low enough to resist against attacks that use correlation analysis.

Another correlation coefficient that is important is correlation between original image and encrypted image. It is calculated by this formula:

$$r = \frac{\sum_i \sum_j (P_{ij} - \bar{P})(E_{ij} - \bar{E})}{\sqrt{(\sum_i \sum_j (P_{ij} - \bar{P}))(\sum_i \sum_j (E_{ij} - \bar{E}))}} \quad (19)$$

Where P and E are original and encrypted image respectively. Also $\bar{p}$ is the mean of P elements and is calculated by below formula:

$$\bar{P} = \frac{1}{m \times n} \sum_{i,j} P_{i,j} \quad (20)$$

With respect to these formulas, below results are gotten for method of [11] and our method:

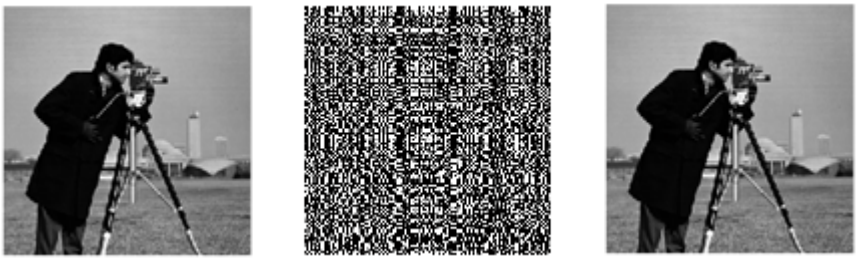
Table 4: correlation coefficient between original and encrypted image.

algorithm	correlation coefficient
Method of [11]	0.0416
Proposed method	0.0150

According to this table, our proposed method decreases this coefficient against method of [11]. So it acts better than [11] in correlation analysis. It is another improvement by our method.

4.6. Sensitivity to encryption key

An image encryption algorithm should be sensitive to encryption key. It means if we encrypt image by K_1 and decrypt by a different key (K_2), decryption should be unsuccessful. It will show that encrypted image can be decrypted only and only by encryption key. Our simulations show proposed method is sensitive to key. Figure 2 proves this and shows decryption with correct and incorrect keys.



a b c

Fig.2. Image encryption and decryption using the proposed method. (a) Original image, (b) Decrypted image with incorrect key, (c) Decrypted image with correct key.

5. Conclusion

In this paper we use 2-D MPDFRFT in a new form to encrypt image. We improved the method of [11] that was presented in December of 2009. [11] used 2-D MPDFRFT for encryption that each of MPDFRFT is created by (10). $\lambda_k^{a_k}$ on its method were made by (5). We completely changed $\lambda_k^{a_k}$ and defined it $(\lambda_k)^{a_k}$. This definition is very sampler and randomization property is provided better by this. As our simulations show, all image encryption criteria are provided. Also speed of algorithm, percentage of fixed points and correlation coefficient between original image and decrypted image are improved in comparison with the method of [11].

6. References

[1] S. C. Pei and M. H. Yeh, “Improved discrete fractional Fourier transform, “Opt. Lett., vol. 22, pp. 1047–1049, 1997.

[2] C. Candan, M. A. Kutay, and H. M. Ozaktas, “The discrete fractional Fourier transform,” IEEE Trans. Signal Process., vol. 48, no. 5, pp.1329–1337, May 2000.

[3] S. C. Pei and W. L. Hsue, “The multiple-parameter discrete fractional Fourier transform,” IEEE Signal Process. Lette., vol. 13, no. 6, pp. 329–332, Jun. 2006.

[4] B. W. Dickinson and K. Steiglitz, “Eigenvectors and functions of the discrete Fourier transform,” IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-30, pp. 25–31, Jan. 1982.

[5] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, the Fractional Fourier Transform with Applications in Optics and Signal Processing. NewYork: Wiley,2000.

[6] L. B. Almeida, “The fractional Fourier transform and time-frequency representations,” IEEE Trans. Signal Process, vol. 42, no. 11, pp. 3084–3091, Nov. 1994.

[7] Z. Liu and S. Liu, “Randomization of the Fourier transform,” Opt. Lett., vol. 32, pp. 478–480, 2007.

[8] J. H. McClellan and T. W. Parks, “Eigenvalue and eigenvector decomposition of the discrete Fourier transform,” IEEE Trans. Audio. Electroacoust., vol. AU-20, no. 1, pp. 66–74, Mar. 1972.

[9] W. F. Trench, “Characterization and properties of matrices with generalized symmetry or skew symmetry,” Lin. Alg. Appl., vol. 377, pp. 207–218, 2004.

[10] S. C. Pei, J. J. Ding, W. L. Hsue, and K. W. Chang, “Generalized commuting matrices and their eigenvectors for DFTs, offset DFTs, and other periodic operations,” IEEE Trans. Signal Process., vol. 56, no. 9, pp. 3891–3904, Aug. 2008.

[11] Soo-Chang Pei, Wen-Liang Hsue, “Random Discrete Fractional Fourier Transform”, IEEE SIGNAL PROCESSING LETTERS, VOL. 16, NO. 12, pp. 1015-1018, Dec. 2000.