Optics and Lasers in Engineering 85 (2016) 72–83

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Novel permutation measures for image encryption algorithms

Salwa K. Abd-El-Hafiz [a], Sherif H. AbdElHaleem [a], Ahmed G. Radwan [a,b,*]

[a] Engineering Mathematics Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt
[b] NISC Research Center, Nile University, Giza 12588, Egypt

ARTICLE INFO

Article history:
Received 16 December 2015
Received in revised form
5 April 2016
Accepted 27 April 2016

Keywords:
Chaos
Chess horse movement
Image encryption
Pixel permutations

ABSTRACT

This paper proposes two measures for the evaluation of permutation techniques used in image encryption. First, a general mathematical framework for describing the permutation phase used in image encryption is presented. Using this framework, six different permutation techniques, based on chaotic and non-chaotic generators, are described. The two new measures are, then, introduced to evaluate the effectiveness of permutation techniques. These measures are (1) Percentage of Adjacent Pixels Count (PAPC) and (2) Distance Between Adjacent Pixels (DBAP). The proposed measures are used to evaluate and compare the six permutation techniques in different scenarios. The permutation techniques are applied on several standard images and the resulting scrambled images are analyzed. Moreover, the new measures are used to compare the permutation algorithms on different matrix sizes irrespective of the actual parameters used in each algorithm. The analysis results show that the proposed measures are good indicators of the effectiveness of the permutation technique.

1. Introduction

In recent years, image encryption has been an active research area because of the demand on securing digital images over the internet. A typical image encryption scheme consists of two main substitution and permutation phases to accomplish Shannon's confusion and diffusion properties [1]. While the permutations phase changes the pixels locations, the substitutions phase changes the pixels values. The substitutions phase can be performed using a pseudo random number generator based on continuous or discrete chaotic systems (e.g., [2–5]). Substitutions can also utilize non-chaotic generators such as fractal shapes [6] or a combination of both chaotic and non-chaotic techniques as in the combined DNA-chaos techniques (e.g., [7–9]). The permutations phase can also be performed using various chaotic and non-chaotic techniques. For instance, permutations can be based on discrete chaotic maps such as the logistic or Arnold's map (e.g., [8,10]). Moreover, permutations can be based on continuous attractors such as the Lorenz attractor or Chen's hyper-chaotic system (e.g., [9,10]). Non-chaotic permutations can use a multitude of methods such as the chess-based horse movement [10] or the trajectory of a water wave movement [11].

Due to the high sensitivity of chaotic systems to parameters and initial conditions, chaos based algorithms are developed and studied for encryption systems, in general, and for the permutation phase in specific. For example, [5] randomly shuffles the image pixels using the 2D Chirikov map, which is an invertible area preserving chaotic map. In [7], permutations are performed based on sorting pseudorandom data coming from a chaotic map. In [12], the image is broken down into 8×8 sub-blocks and then row and column-wise rotations are performed based on a chaotic key. The chaotic keys used in the permutations as well as substitutions are generated based on four different chaotic maps; logistic, tent, quadratic and Bernoulli. In [13], a large permutation with the same size as the plain image is used to shuffle the positions of image pixels totally. An effective method is presented to construct the large permutation by combining several small permutations, where small permutations are directly generated using a chaotic map. Alternatively, [14] presents two permutation methods based on a perturbed piecewise linear chaotic map: the cyclic shift bit permutation method and a bit permutation method. The former can be a permutation of bits, bytes, or a set of bytes, and the latter is applied on 8 bits whose positions are also controlled by chaos.

In addition, Arnold's cat map is one of the most studied invertible chaotic maps, particularly, in image encryption, watermarking and steganographic algorithms (refer, for instance, to [15–22]). In [15] and [16], the 2D Arnold's cat map is generalized to a 3D map and, then, it is used in permutations. Rather than using the ordinary 2D Arnold's cat map, [17] uses the reverse map in performing pixel permutations. In [18], the 2D Arnold's cat map is generalized to take real parameters rather than integer ones. This generalized map is, then, utilized to generate one chaotic orbit,

* Corresponding author at: Engineering Mathematics and Physics Dept., Faculty of Engineering, Cairo University, Giza, 12613, Egypt.
E-mail addresses: salwa@computer.org (S.K. Abd-El-Hafiz), sherifHamdyNet@hotmail.com (S.H. AbdElHaleem), agradwan@ieee.org (A.G. Radwan).

http://dx.doi.org/10.1016/j.optlaseng.2016.04.023
0143-8166/

which is used to get two-index order sequences for the permutation of image pixel positions. Furthermore, [19] uses a dynamic random growth technique to permute image pixels using another form of a generalized Arnold's cat map.

The logistic map is also commonly used in image encryption (refer, for instance, to [5,8,12,13,15,17,19,21–27]). The 2D logistic map is used in [8] to produce two sequences, which are sorted and, then, used to create a permutation matrix. In [15], a sequence of numbers is generated using the logistic map to control the iterations of the 3D cat map used in the permutations. Similarly, [22] generates two random number sequences by the logistic map to define the two parameters of the 2D Arnold's cat map. Ref. [23] makes use of the chaotic property of the logistic map to perform both pixel permutations and substitutions. The adjacent pixels of an image in a row are XORed together and, based on a chaotic key, the pixels are scrambled. The same scrambling is performed for the image columns and the combination of both row and column scramblings will form the permuted image. In addition, [24] generates a large pseudorandom permutation matrix, which is combinatorially generated from small permutation matrices based on the logistic map. In [25], a coupled map lattice, based on the logistic map, is used to generate a sequence of pseudorandom numbers. Those numbers are then used in cyclic right and cyclic up operations on each row and column, respectively.

Other research uses the continuous Chen hype-chaotic system or the Lorenz attractor in the permutations phase (e.g., [9,28–30]). The Chen hyper-chaotic system is used in [9] to produce four sequences and two sequences are sorted to form the permutation matrix. In [28], the image is viewed as a 3D cube and each of the real valued coordinates of the cube are considered as initial values for the Lorenz attractor. After some number of iterations, which is decided by the key, each coordinate of the normalized cube ends up at another three dimensional real valued coordinate that is used to fill up a permutations matrix. Instead of generating a permutations matrix, [29] and [30] produce an encryption matrix where one iteration of Lorenz system generates three variables that are post-processed such that two of them are used for position permutation and the third is used for grayscale substitution.

It should also be mentioned that most block-cipher image encryption schemes, based on chaos theory, have independent modules for the permutation and substitution processes. However, some systems combine the permutation and substitution processes into one phase. For instance, [31] embed the permutation mechanism into substitutions by applying a 3D Baker map based substitution algorithm. Moreover, [27] presents an integrated permutation–substitution mechanism, which combines the two-dimensional logistic chaos with four-dimensional hyper-chaos.

On the other hand, non-chaotic methods are also commonly used in implementing the permutations phase. For instance, [32] uses the Latin squares algorithm to design a new 2D substitution–permutation network. Ref. [33] divides the image into non-overlapping blocks and each block is scrambled using a zigzag-like algorithm. Furthermore, [34] divides the image into a set of k-bit vectors; each of these vectors is substituted and then permuted by circularly right rotating its bits. Alternatively, [35] divides the image into non-overlapping blocks and for each encryption round the size of the block changes according to the round key. Within the same block, permutations are performed using a zigzag like algorithm. After performing the value substitutions, row and column transformations are performed over the processed block. Ref. [11] imitates the trajectory of water wave movement to scramble the image in the horizontal, vertical and diagonal directions. Moreover, self-adaptive permutations are utilized in [36] by using the pixel values of two color components to control swapping the rows and columns of the left third component. While [37] uses a chess-based algorithm, which simulates the horse movement, to perform the permutation process, [38] transforms the image into a one-dimensional array of pixels and uses a Gray-code based permutation technique.

In summary, permutations are performed using chaotic and non-chaotic algorithms in many different techniques. Therefore, the main objective of this paper is to present two new measures for the evaluation and comparison of different permutation techniques. To demonstrate the ability of the proposed measures to evaluate different permutation strategies, six permutation techniques with dissimilar approaches and varying outcomes are selected. The two measures are used to compare those six different permutation techniques, which utilize chaotic and non-chaotic generators.

Section 2 of this paper describes the six different permutation techniques in a unified mathematical framework. While Section 3 introduces the two new permutation measures, Section 4 utilizes them to compare the six permutation algorithms using different standard images and different analysis techniques. In addition, Section 4 analyzes the complexities and execution times of the proposed measures. Finally, Section 5 provides conclusions and future work directions.

2. Different permutation techniques

The permutation phase is a process in which each pixel location is changed to another location. This process must be a one-to-one correspondence to be able to recover the image again without distortion. The permutation phase can be represented using a matrix where each matrix element holds the new position of the pixel. A permutation matrix T of size $M\times N$ is defined as in (1). Assuming that the image pixels are processed from to top to bottom and from left to right, the new location of each pixel is calculated using (2).

$$T=\left\{\begin{array}{c} T_{ij};\, T_{ij}\in\{1,\,2,\,\ldots,M\times N\};\, T_{ij} \text{ are distinct;} \\ i\in\{1,\,\ldots,M\},\, j\in\{1,\,\ldots,N\} \end{array}\right\}. \tag{1}$$

$$Col_{new}=div\left(T_{ij}-1,M\right)+1 \tag{2a}$$

$$Row_{new}=mod\left(T_{ij}-1,M\right)+1, \tag{2b}$$

where $i\in\{1,\ldots,M\}$ and $j\in\{1,\ldots,N\}$ are the row and column indices, respectively. As an example, Fig. 1(a) shows the pixel processing indices in a block of size 4×4, Fig. 1(b) shows the permutation matrix T and Fig. 1(c) shows the new location of each pixel. Let $i=1$ and $j=1$, then T_{ij}=8 and using (2) Col_{new}=2 and Row_{new}=4 which means that the pixel at location (1, 1) in the old block will be placed at location (4, 2) in the new block. In the following subsections, six different techniques for generating a permutation matrix in the form of (1) are presented.

2.1. Matrix generation using discrete chaos

In this technique, the logistic map is used to generate a sequence of values and any discrete chaotic map can similarly be used. These values are then sorted in ascending order and the new index for each value, in the sorted sequence, is used to fill up the permutation matrix. The conventional logistic map with parameter λ is defined as follows:

$$r_{n+1}=\lambda r_n\left(1-r_n\right). \tag{3}$$

For a matrix T of size $M\times N$, the discrete chaotic system should