Compression-friendly Image Encryption Algorithm Based on Order Relation

Ganzorig Gankhuyag, Yoonsik Choe

Department of Electrical & Electronics Engineering, Yonsei University, South Korea
gnzrg25@yonsei.ac.kr, yschoe@yonsei.ac.kr

Abstract

In this paper, we introduce an image encryption algorithm that can be used in combination with compression algorithms. Existing encryption algorithms focus on either encryption strength or speed without compression, whereas the proposed algorithm improves compression efficiency while ensuring security. Our encryption algorithm decomposes images into pixel values and pixel intensity subsets, and computes the order of permutations. An encrypted image becomes unpredictable after permutation. Order permutation reduces the discontinuity between signals in an image, increasing compression efficiency. The experimental results show that the security strength of the proposed algorithm is similar to that of existing algorithms. Additionally, we tested the algorithm on the JPEG and the JPEG2000 with variable compression ratios. Compared to existing methods applied without encryption, the proposed algorithm significantly increases PSNR and SSIM values.

Keywords: Image encryption, Order relation, Compression-friendly encryption, Order permutation, Image scrambling

1 Introduction

With the rapid changes in multimedia and communication technology, large amounts of digital image data are now being transmitted over the Internet [1-2]. In general, these transmissions use public networks with limited bandwidth, and are not sufficiently secure to transmit sensitive data, such as digital images of army emplacement, satellite images of military bases, and medical images of patient data [3]. To maintain information security and prevent illegal access by unauthorized users, image encryption algorithms have become a vital area of research [4].

Various methods in chaotic map-based encryption algorithms have been suggested over the past decade [5-8]. Liu and Wang proposed the one-time key cryptosystem algorithm based on dual chaotic map [9]. Hongjun et al. proposed a bit level permutation method that can change position of pixel and its value [10]. Some of researchers used a DNA complementary rule and chaotic map to encrypt the image [11-12]. Mitra et al. proposed a random permutation encryption algorithm based on bit permutation, pixel permutation, and a simple structure to perform block permutation [13]. Manimurugan and Porkumaran recommended an encryption algorithm that applies a block pixel-sorting algorithm to compress and permute both column and row data [14]. Mahdieh et al. proposed an encryption algorithm utilizing a logistic chaos system that permutes the horizontal and vertical rows of the image. Then, a block encryption algorithm encrypts the output image [15]. Existing encryption algorithms use permutation and confusion stages. Some iteratively utilize the two stages, while others apply only a scrambling stage, depending on the application. Many researchers have proposed scrambling image encryption algorithms [16-22]. Guan et al. presented an image encryption scheme that shuffles the positions, and changes the grey values of image pixels [21]. Kekre et al. proposed a perfect shuffle using different factors of image size for image scrambling in which certain number of iterations is required to obtain the original image [22]. The previous approaches are able to improve security via pixel correlation elimination technique with only permutation or permutation plus confusion in the original image in various methods, depending on the purpose. The permutation method changes the positions of image pixels but does not affect their values. In contrast, the confusion method changes image pixel values and random changes of pixel spread throughout the image, increasing the compression efficiency loss in lossy compression. [23]. In other words, an image compression method that emphasizes correlation between pixels have an adverse impact on performance due to random pixels with no correlation which handles the encryption. Therefore, an encryption permutation method with no impact on the image compression and also maintaining the security of the image are much needed.

The compressive sensing (CS) based image encryption scheme proposed a method that can perform encryption and compression at the same time [24-25].

[*] Corresponding Author: Yoonsik Choe; E-mail: yschoe@yonsei.ac.kr

DOI: 10.3966/160792642020072104011

The main idea of CS based encryption algorithm is to transform the original image into a set of k-sparse domain such as frequency domain, then use the measurement matrix in compression and encryption process. However, CS based encryption algorithm is required to solve complex optimization equations to decryption and decompression in order to get the original image, as it cannot be extended to standard compression algorithms.

In this paper, we present a new encryption algorithm, more efficient with lossy and lossless compression algorithm, and capable with well-known compression standard JPEG and JPEG200. The proposed algorithm is based on the order relation theory [26] which maximize the pixel correlation through correlated set ordering method to provide security and improve the compression efficiency simultaneously. The proposed image encryption method performs separately from the compression process; it is compatible with the conventional compression algorithms. The rest of the paper is organized as follows. Section 2 describes the order relation encryption algorithm, and Section 3 analyzes the security strength and compression friendliness of the proposed algorithm, comparing it with previous works. Finally, Section 4 provides the conclusions of this paper.

2 Proposed Image Encryption Method

In this section, we illustrate the proposed encryption method. First, we provide a brief description of the order relation set theory and information partitioning method applied to the digital image. Then, we describe the permutation method which includes divide subpartition method and permute subpartition method.

2.1 Order Relation Applied to a Digital Image

Set theory. Let *A* and *B* be two sets of natural values. A relation between *A* and *B* is a subset of $A \times B$. In other words, relation *R* is a subset of ordered pairs of *(a, b)* such that $a \in A$ and $b \in B$, as in (1).

$$R = \{(a, b) : a \in A, b \in B\} \tag{1}$$

where the domain of a relation *R* from *A* and *B* is the set of all first elements of the ordered pairs, and the range of *R* is the set of all second elements [26].

A digital image consists of coherent pixels. For an 8-bit grayscale image, the range value of image pixel intensity is 0 to 255. Let *S*={0,1,…,255} and *Q* be any sequence consisting of elements of *S*, as in (2).

$$Q = \{q_i\ q_i \in S, i \in N\} \tag{2}$$

where $i \neq j$ does not imply $q_i \neq q_j$. Let *P* is pixel sets of location. Suppose that the cardinalities of *P* and *Q* are the same; then, a relation *I* between *P* and *Q*, the elements of which are binary (or ternary) relations such as *(p, q)*, is expressed a digital image, as defined in (3).

$$I = \{(p, q) \mid p \in P, q \in Q\} \tag{3}$$

where *P* is the set of pixel, and *Q* is the intensity set of pixel.

Information partitioning. A partition of a set *x* is a set of non-empty subsets of *X* such that every element *x* in *X* is in exactly one of these subsets [27]. Therefore, a family of sets *T* is a partition of *X* when the following conditions hold:

(1) $T_i \neq 0$ for $i = 1, \ldots, k$

(2) $U_{i=1}^{k} T_i = X$

(3) $T_i \bigcup T_j$ if $i \neq j$

In fact, for any equivalence relation on a set *X*, a set of its equivalence classes form a partition of X.

The relation based image data is divided an image into two sets: the set of pixels *P* and the set of intensities *Q*. However, the uncertainty of *P* is zero because the set *P* is a well-ordered set with a lexicographic order. Let signal *I* be a relation between *P* and *Q*, where *P*={1,…,k} and *Q*={1,…,n}. A relation *I* is $| I_i - I_{i+1} | = 0$. Then, the relation *I* is an equivalence relation. The set partitioned by the equivalence relation results in non-overlapped equivalence sub classes [28].

2.2 Image Permutation Method

The proposed method is separately applied to the image in horizontal and vertical direction permutation. The proposed algorithm assumes that the original images have the dimensions $M \times N$. In case of horizontal permutation, the dimension of row data is $1 \times N$, and the value of x_{iN} can be any positive integer that satisfies $x_{iN} \in (1, \ldots, M)$, where *M* is the total row number. The row number is increased by one until end of the image in horizontal permutation. We applied the row data x_{iN} in equation (3) is defined in (5).

$$I = \{(p, q) \mid p \in N, q \in x\} \tag{5}$$

Where the *q* set is the intensity values of x_{iN} and the *p* set is corresponding pixel location of x_{iN}. The vertical permutation is performed similarly to horizontal permutation. In vertical permutation, column data x_{Mj} can be any positive integer that satisfies $x_{Mj} \in (1, \ldots, N)$, where N is the total column number. The column number is increased by one until end of the image in vertical permutation.

The permutation process involves two steps in each direction: divide the data into *I* subpartition then permute the *I* subpartition, as shown in Figure 1. In the first step, we split the input pixels into subpartition, but it is not just a random data division into subpartition. The data with correlated pixels are combined into one subpartition. Hence, the correlated pixel range is defined by data partition boundary. The partition boundary of data has to be adaptively decided for data instead of uniformly decided. In this paper, the partition boundary is calculated using the well-known

Lloyd algorithm which specialized for dividing data into optimal partitions [29]. After that, input pixels are divided into subpartition, so correlated pixels are included into same subpartition keeping the correlation between pixels in subpartition.

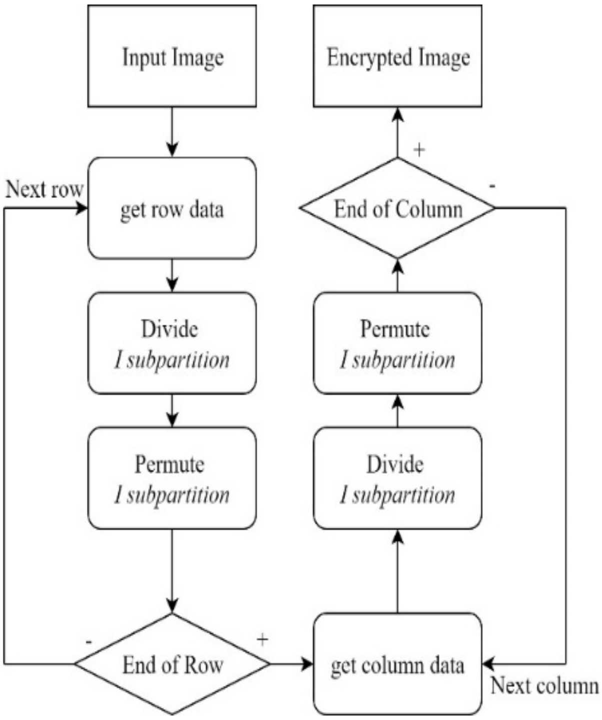


Figure 1. Block diagram of the proposed algorithm

And second step, we permute the *I* subpartition. Though the permutation method is applied in a random way, we make the descending order based on the representative value of such subpartition. Thus, data that is correlated within a subpartition also obtain correlation with adjacent subpartitions by permutation. It eventually makes the data unpredictable becoming as one form of the encryption. In addition, the correlation of data is greatly improved, which improves compression efficiency.

2.2.1 Divide Subpartition

The frequencies of the row or column data pixel values are combined to form *I* subpartition. To combine subpartition, we firstly need to determine the partition boundary based on Lloyd's algorithm and then decide the corresponding pixels in each *I* subpartition.

2.2.1.1 Decide Partition Boundary

The partition boundary decision process starts with selecting arbitrary representative value for partition then repeating following steps:

Step 1: Assign each pixel to the new partition corresponding to its nearest representative value.

Step 2: The assignment of pixel to partitions, compute new representative value of partition.

The algorithm stops when non-overlapped optimal partitions are identified with their corresponding boundary values. Additionally, each partition has their corresponding threshold value which low threshold th_s^l and a high threshold th_s^h , where *s* is the partition number.

2.2.1.2 Decide Pixel in Subpartition

After determining the boundary of each subpartition, we decide corresponding pixel to each subpartition. To determine each *I* subpartition, the algorithm starts with the first pixel x_{ij}.

Step 1: The pixel x_{ij} compares with threshold values of the partition boundaries until the identified relevant partition increases partition number.

$$th_s^l \le x_{ij} < th_s^h \tag{6}$$

Step 2: If a relevant partition is determined, then the pixel value x_{ij} is added to the q_t set and subpartition number *t* added to the p_t, where *t* is the *I* subpartition number.

Step 3: After allocation, the current pixel x_{ij} to subpartition, it repeats the Step 1 to find the relevant partition for next pixel x_{ij+1}.

Step 4: If x_{ij+1} has the same partition as x_{ij}, then x_{ij+1} is added to q_t and subpartition number added to p_t of subpartition I_t. In this case, cardinality of I_t is increased by 1.

Step 5: If x_{ij+1} does not belong to the same partition, then x_{ij+1} is allocated to a new subpartition *I*, and the number of set *t* increased by 1.

These steps are repeated until all pixels in a row are divided into subpartitions. In Figure 2 shows that illustrated example of divide subpartition step. Figure 2(a) shows the original data, x-axis shows the pixel location and y-axis shows the pixel intensity. This example illustrates 11 different pixel data. Figure 2(b) shows the divided *I* subpartition data of original data, x-axis shows the value of *p* set, and y-axis shows the value of *q* set of corresponding *I* subpartition, total subpartition number is 4, and corresponding value of q_t and p_t are given in x axis and y axis.

2.2.2 Permute Subpartition

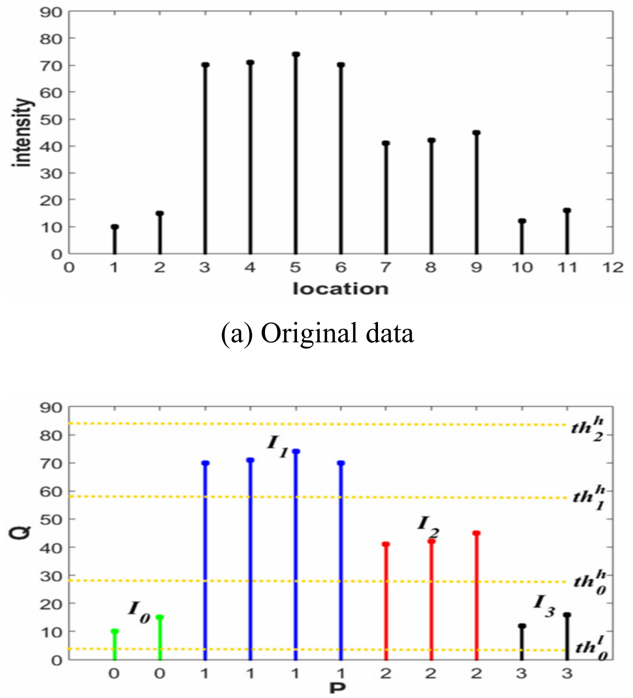
The independent partitions of *I* were created in the divide subpartition step. Next, we permute each *I* subpartition by representative value.

Step 1: The representative value is the average value of pixels in each *I* subpartition, as defined in (7).

$$I_t^{representative} = \sum_{i=1}^{k} q_i \tag{7}$$

where *k* is cardinality of corresponding I_t.

Step 2: The subpartition in row is permuted in descending order based on representative value to disrupt the neighboring pixel correlation between the subpartitions.



(a) Original data

(b) Divided *I* subpartitions

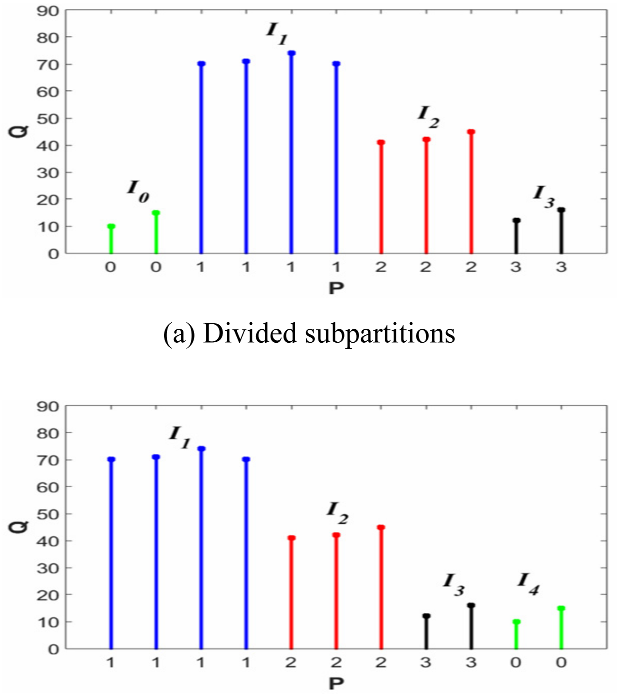
Figure 2. Illustrated example of divide subpartition

Once permuted, the neighboring uncorrelated behaviors between the subpartitions provide a form of encryption. Illustrated example of permutate subpartition step is show in Figure 3. Figure 3(a) shows the divided subpartitions, Figure 3(b) indicates the permuted subpartition based on their representative value. In this example, the discontinuity between I_0, I_1 and I_2, I_3 are eliminated by permutation process. Also, subpartition I_3 and I_0 become neighbor data. Furthermore, the correlation of neighboring pixel is increased by permutation. Because the subpartitions are permuted by descending order, it results in high correlation with adjacent pixels. Figure 5(a) to Figure 5(f) and Figure 5(g) to Figure 5(l) respectively display the original image and the proposed encrypted image.

After horizontal permutation process, vertical permutation process repeats the divide subpartition and permute subpartition step until end of column. The permutation process generates the encryption key in each direction. In Figure 4. the horizontal and vertical permuted example of image is shown. The encryption key is including *I* subpartition's location data p_t and the cardinalities of *k*. For example, the encryption key of Figure 3(b) is {[1, 4], [2, 3], [3, 2], [0, 2]}.

2.3 Image Decryption Method

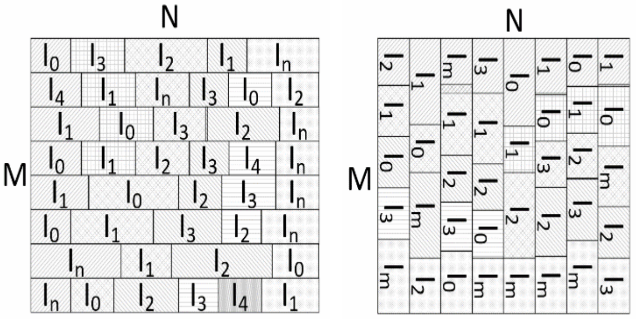
The decryption process requires two different data: the proper encryption key and the encrypted image that generated from the encryption process. The encryption key contains the value of p_t sets and the cardinality value for each p_t corresponding to the rows and columns. The encrypted image contains a set of permuted q_t based on *I* subpartitions in the encryption process. In terms of decryption, it first decrypts the column data and then decrypts the row data as opposed to encryption. For decryption, make each column or row data into an *I* subpartitions containing q_t and p_t sets. Then, re-permute the subpartitions by ascending order based on the p_t value. For example, the decryption process gets the encryption key of illustrated example Figure 3(b) {[1, 4], [2, 3], [3, 2], [0, 2]}, after re-permutation {[0, 2], [1, 4], [2, 3], [3, 2]}, last 2 pixels are shifted to beginning of data.



(a) Divided subpartitions

(b) Permuted subpartitions

Figure 3. Illustrated example of permute subpartition


N
M
I_0 I_3 I_2 I_1 I_n
I_4 I_1 I_n I_3 I_0 I_2
I_1 I_0 I_3 I_2 I_n
I_0 I_1 I_2 I_3 I_4 I_n
I_1 I_0 I_2 I_3 I_n
I_0 I_1 I_3 I_2 I_n
I_n I_1 I_2 I_0
I_n I_0 I_2 I_3 I_4 I_1
N
M
I_2 I_1 I_0 I_3 I_m
I_1 I_0 I_m I_2
I_m I_1 I_2 I_3 I_0
I_3 I_1 I_2 I_0 I_m
I_0 I_1 I_2 I_m
I_1 I_0 I_3 I_2 I_m
I_0 I_1 I_2 I_3 I_m
I_1 I_0 I_m I_2 I_3

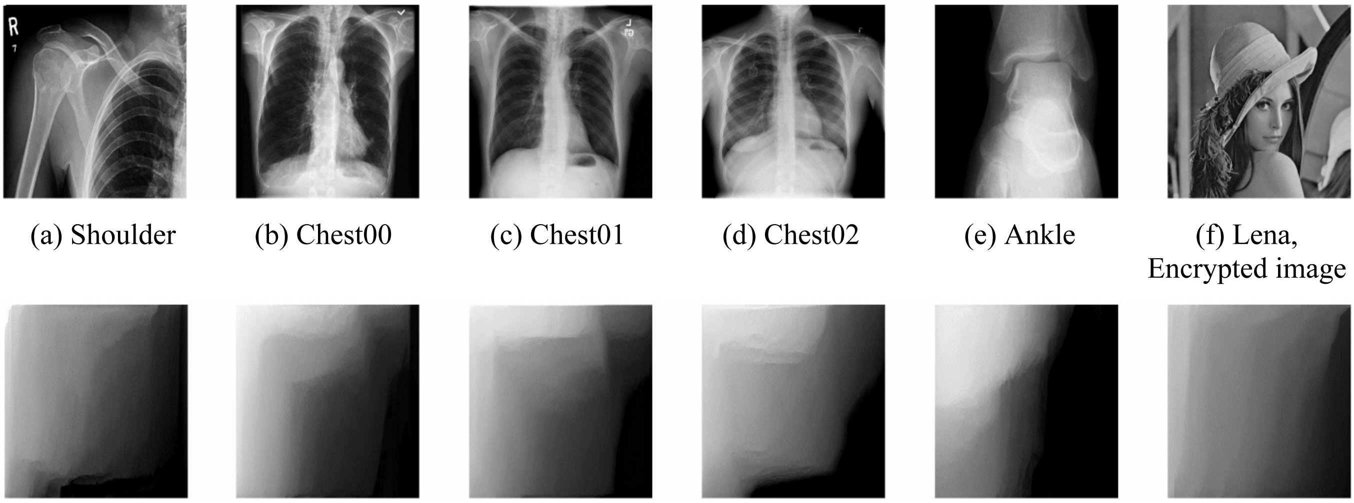
(a) horizontal permuted image

(b) vertical permuted image

Figure 4. Horizontal and vertical permuted image

2.4 Compression and Permuted Subpartition

The main goal of the proposed method is to propose the permutation method of encryption that can improve compression efficiency. For compression efficiency, rather than random pixel permutation, we partition the correlated neighboring pixels into subpartition then permute all subpartitions in horizontal and vertical direction of the image. In the dividing subpartition process, we eliminate the pixel discontinuity of the image. Also, in permute subpartition process, we maximize the correlation of neighboring subpartitions. Thus, by dividing *I* subpartition and permuting *I* subpartition methods, it gets rid of the discontinuities in the image, giving a highly correlated pixels in image. Higher pixel correlation results in more energy concentrated in the DC coefficient and less AC coefficient during the transformation process of image compression algorithm. It helps to reduce quantization error in AC coefficient in quantization process for lossy compression and give more efficiency in entropy coding. In Figure 5(a) to Figure 5 (l) shows the encrypted image, the encrypted image data shows that the adjacent pixel values of encrypted image are very similar. Furthermore, our encrypted images are highly correlated with adjacent pixels, and it improves the compression efficiency.



(a) Shoulder (b) Chest00 (c) Chest01 (d) Chest02 (e) Ankle (f) Lena, Encrypted image

(g) Shoulder (h) Chest00 (j) Chest02 (j) Chest02 (k) Ankle (l) Lena

Figure 5. The original test image and encrypted images using the proposed algorithm

3 Analysis Results

In this section, we evaluate the proposed work in two ways. Firstly, we evaluate the security strength of the proposed encryption with correlation coefficient analysis, key space analysis, mean absolute error analysis. However, in this analysis, we did not take the histogram analysis because the proposed work did not change the pixel values during the encryption process. Second, the compression efficiency of the proposed work is evaluated with the compression-friendly analysis. Our simulation uses six gray-scale images. Five are medical images: Shoulder, Ches00, Chest01, Chest02, and Ankle, shown in Figure 5(a) to Figure 5(e). The other image is Lena shown in Figure 5(f). The image size of all six images is 1024 × 1024 pixels.

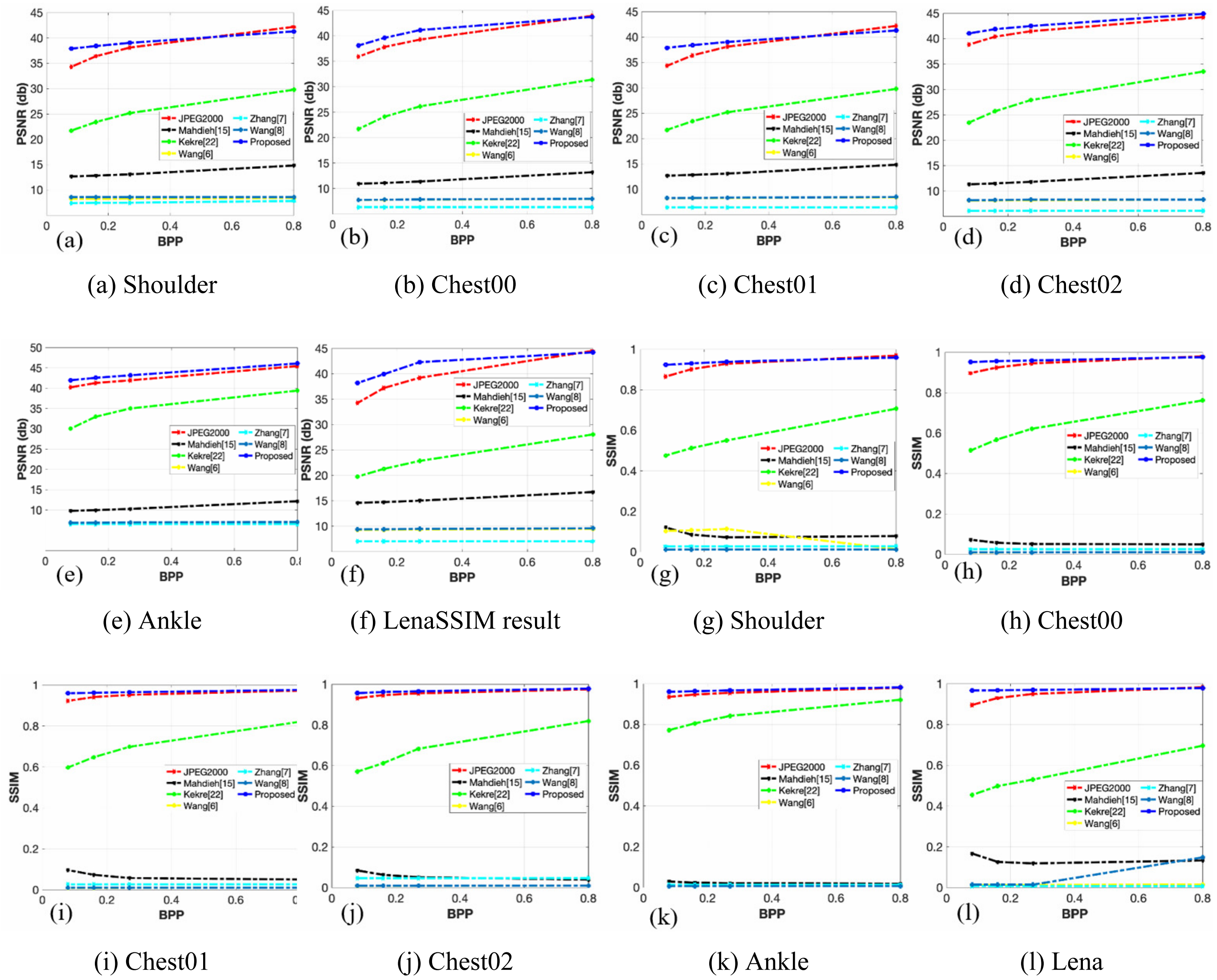
3.1 Correlation Coefficient Analysis

A correlation coefficient provides a measurement of image encryption, and indicates a predictive relationship that can be exploited in practice [5, 14, 30]. This coefficient indicates the strength of the correlation between two variables, and its value ranges from -1 to 1. Values close to 1 or -1 indicate that the variables are strongly correlated, and values close to 0 indicate weak or no correlation.

In this experiment, we compared correlation coefficient analyses of the proposed algorithm with only permutation encryption algorithm: Kekre [22], and permutation plus confusion algorithms: Mahdieh [15], Wang [6], Zhang [7] and Wang [8]. The correlation coefficients between the original and encrypted images are tested by randomly selecting 10,000 pairs of adjacent pixels from each image. Correlation coefficients are calculated as:

$$Cor = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \quad \textbf{(8)}$$

where *x, y* are the pixel values of two adjacent pixels, and $\bar{x}$, $\bar{y}$ are the mean values of the pair images. *N* is the total number of pixel pairs randomly selected from the test images. The correlation coefficients of 6 different pairs are shown in Table 1. For Mahdieh's algorithm, the correlation coefficient of the encrypted image pair #2 is 0.007, and that of the non-encrypted image pair #2 is 0.551.



(a) Shoulder (b) Chest00 (c) Chest01 (d) Chest02

(e) Ankle (f) LenaSSIM result (g) Shoulder (h) Chest00

(i) Chest01 (j) Chest02 (k) Ankle (l) Lena

Figure 6. JPEG2000 result of previous and proposed algorithms with lossy compression

Table 1. Correlation coefficients of previous works and the proposed work

Pair image	Non-encrypted	Encryption algorithm					
		Mahdieh [15]	Kekre [22]	Wang [6]	Zhang [7]	Wang [8]	Proposed work
1. Ankle : Lena	0.085	0.005	0.079	0.011	0.035	0.111	0.898
2. Chest00 : Ankle	0.551	0.007	0.552	0.014	0.163	0.438	0.950
3. Lena : Chest00	-0.010	-0.002	0.025	0.019	0.015	0.033	0.952
4. Chest02 : Chest01	0.701	0.689	0.704	0.004	0.136	0.442	0.954
5. Shoulder : Chest02	0.050	0.028	0.042	0.010	0.057	0.040	0.967
6. Chest01 : Shoulder	0.007	0.017	0.009	0.004	0.024	0.083	0.946
Average	0.231	0.124	0.235	0.010	0.072	0.191	0.945

This result shows that encryption by permutation plus confusion algorithm which Lakhami et al. [15], Wang et al. [6], Zhang and Wang [7], Wang et al. [8] are randomizing the original images into an encrypted image where encrypted images are not correlated with each other. For Kekre's algorithm, the correlation coefficient of the encrypted image pair #2 is 0.552, exhibiting a down sampling effect. In contrast, the correlation coefficient of the proposed algorithm value is 0.950, which means that encrypted image of proposed algorithm is similar to other encrypted image resulting from ordering *I* subpartitions in permutation stage. The average of correlation coefficient value is 0.944. Furthermore, with the proposed algorithm encryption, the all encrypted image is highly correlated with other encrypted images. The proposed algorithm changes the subpartitions locations, making the original image unpredictable from the encrypted image.

3.2 Key Space Analysis

The key space analysis is summarized in the following sections. The encryption key is determined by the I subpartitions that were divided by Lloyd's algorithm. The number of I subpartitions varies for each row and column. For an $M \times N$ pixel image size, the total number of I subpartition is given by (9).

$$\text{total\_I\_subpartition} = \left(\sum_{i=1}^{M} \sum_{j=1}^{s} I_{ij} \right) + \left(\sum_{i=1}^{N} \sum_{j=1}^{K} I_{ij} \right) \quad \textbf{(9)}$$

Where S and K are the maximum number of I subpartitions for each row and column, respectively. Let 2^b bits represent the total number of I subpartitions, where, for a good level of security, the key space should be larger than 2^{100} [31-32]. For example, for our proposed encryption algorithm and the test image Ankle, the total number of I subpartitions is 2^{298433}, much larger than 2^{100}. As the image size increases, the number of I subpartitions increases greatly. Therefore, the number of possible values for the encryption key also increases greatly. This makes unauthorized decryption without the proper encryption key more difficult.

3.3 MAE Analysis

The measurement of mean absolute error (MAE) indicates a difference between the original image and the encrypted image and is defined as [15, 22]:

$$MAE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} | O(i,j) - E(i,j) |}{M \times N} \quad \textbf{(10)}$$

where $O(i, j)$ is the pixel value of the original image, and $E(i, j)$ is the pixel value of the encrypted image. The calculated MAE values for the previous and proposed algorithms are listed in Table 2. The MAE value of the Chest00 image is 60.41, indicating that there is no similarity between the original image and the encrypted image. The average value of proposed work's MAE is higher than Kekre's work by 25.19 and similar to the other works. In terms of MAE analysis, the prosed work provides similar security strength with permutation plus confusion algorithms. Furthermore, the original image unpredictable from the encrypted image, there is no similarities between original image and encrypted images.

Table 2. MAE values of previous and proposed algorithms

Test Image	Lakhami et al. [15]	Kekre et al. [22]	Wang et al. [6]	Zhang and Wang [7]	Wang et al. [8]	Proposed work
1. Shoulder	35.28	15.22	41.21	41.55	41.79	36.17
2. Chest00	49.55	28.31	42.61	42.15	42.53	60.41
3. Chest01	39.85	16.26	41.91	41.59	41.85	37.65
4. Chest02	41.82	14.36	37.48	37.34	37.45	41.99
5. Ankle	49.01	8.12	44.16	43.67	44.06	55.60
6. Lena	37.52	17.10	49.68	49.13	49.51	30.68
Average	42.17	16.56	42.84	42.57	42.87	43.75

3.4 Compression-friendly Analysis

This section describes the compression-friendly analysis. The proposed scheme is implemented as an extension of JPEG and JPEG2000 compression standard [33], as shown in Figure 7.

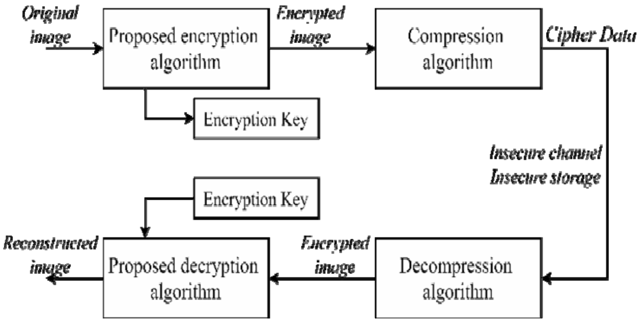


Figure 7. Scheme of the compression-friendly analysis

For the compression-friendly analysis, we varied the compression ratio for JPEG2000 from 10 to 100 and JPEG from 9 to 42 for each test image. In addition, we used two metrics for visual degradation and compression friendliness. The peak signal to noise ratio (PSNR) is defined in (11).

$$MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [O(i,j) - E(i,j)]^2}{M \times N} \quad \textbf{(11)}$$

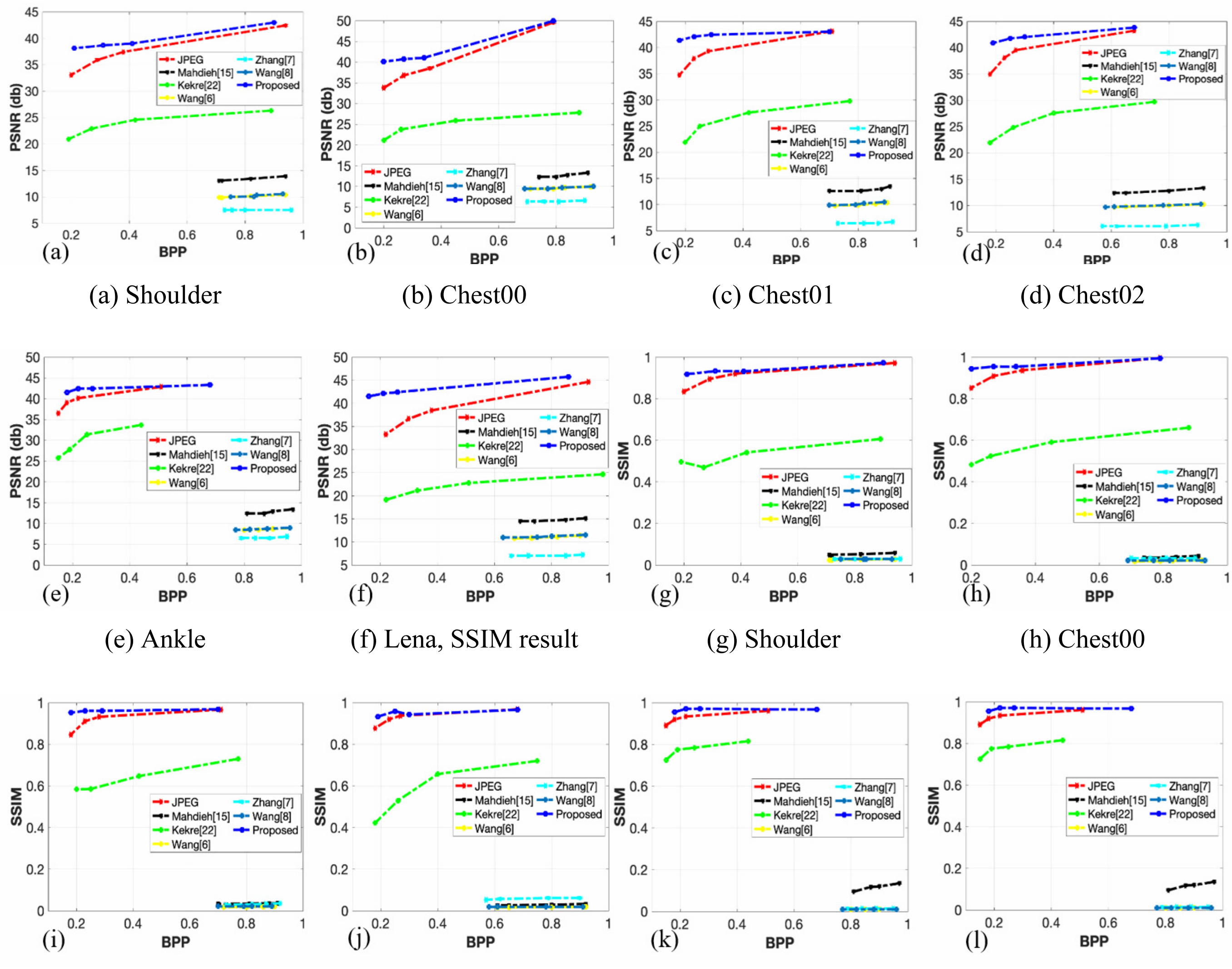
$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

where MAX is the maximum possible pixel value of the image. When test image pixels are represented as 8 bits per pixel, the MAX is 255.

The second metric is the structural similarity index method (SSIM), which measures the similarity between the original image and the distorted image. The result of SSIM index is between -1 and 1, and 1 means two images are indicating perfect structural similarity. A value of 0 indicates no structural similarity.

In Figure 6(a) to Figure 6(e), the y-axis is the PSNR value of test image compression using the JPEG2000 algorithm. The x-axis indicates the compression ration by BPP (bit per pixel). The non-encrypted Ankle images has a PSNR value of 41.94 dB when compressed with compression ratio 30 (0.27 bpp). If we apply permutation plus confusion algorithm, the PSNR value becomes under 10 dB, which is an unacceptable value in image compression. Although these algorithms achieve security, it is inconvenient with lossy compression because of the permutation process. After the permutation process, no continuity in pixel values remains in the encrypted image. The existence of discontinuity increases the approximation error originating from compression applied in the spatial domain. In various transform-based lossy compression methods, the performance of compression is determined by the continuity of estimation during the process. The PSNR value of Kekre's approach is 35.01 dB, illustrating compression loss. In contrast, the PSNR value of the proposed algorithm is 43.18 dB, greater than that of the original compressed image. This compression efficiency improvement is achieved by the permutation methods of our proposed algorithm. The resulting reduction in pixel value discontinuities eliminates the estimation (approximation) error and thereby enhances the compression efficiency. The improvement varies depending on the image and compression ratio. The approximation error is generated more in high compression ratio to compress more. Therefore, proposed work is to increase the correlation of image and it is more efficient in high compression ratio.

In Figure 6(g) to Figure 6(l), the y-axis is the SSIM value of the test images applied to JPEG2000. The experimental results show that the proposed encryption algorithm increases the structural similarity value.



(a) Shoulder

(b) Chest00

(c) Chest01

(d) Chest02

(e) Ankle

(f) Lena, SSIM result

(g) Shoulder

(h) Chest00

(i) Chest01

(j) Chest02

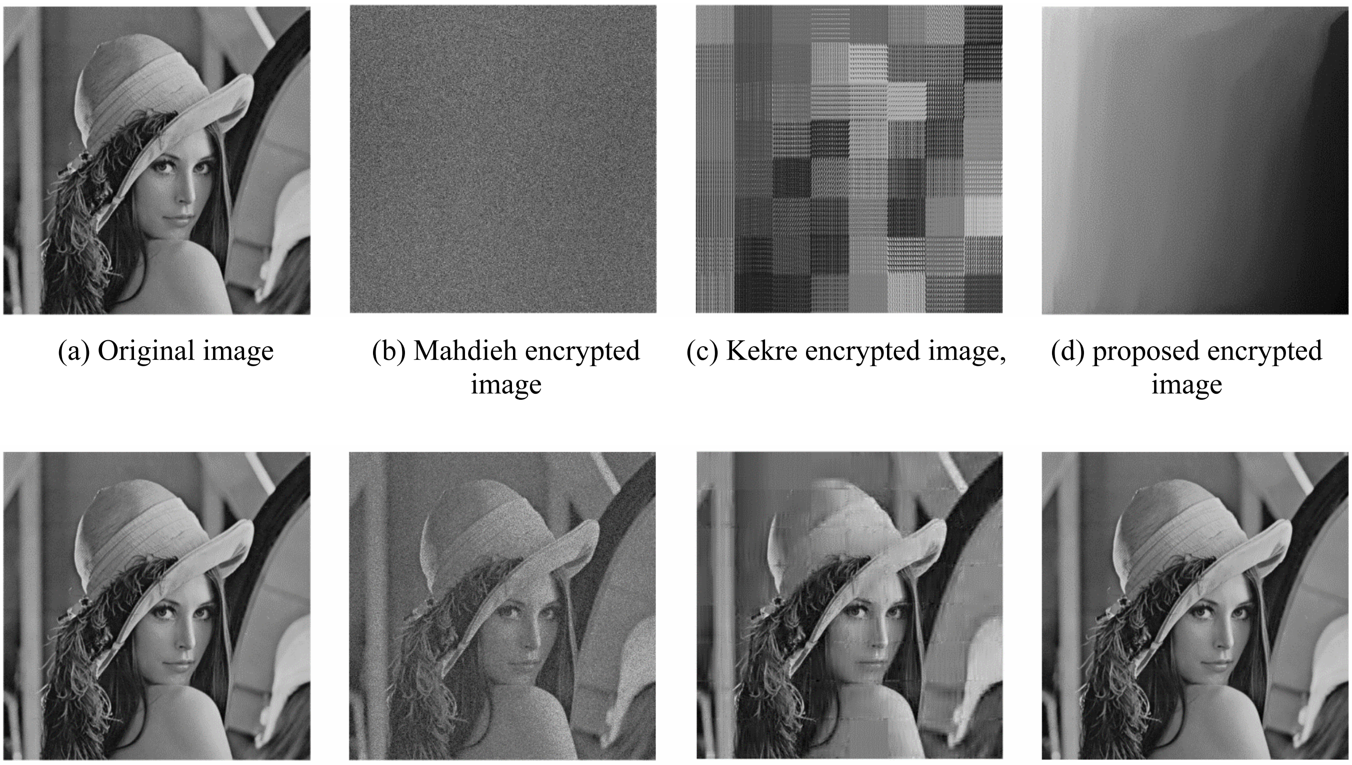
(k) Ankle

(l) Lena

Figure 8. JPEG result of previous and proposed algorithms with lossy compression

As seen in the result the SSIM difference between Ankle original and Ankle proposed are 0.0018 in compression ratio 10 (0.8 bpp).

However, difference is increased by 0.0252 when compression ratio is 100 (0.08 bpp). In point of structural information, proposed work is to save more structural information loss in high compression ratio. The example of reconstruction image with JPEG2000 is shown in Figure 9.



(a) Original image

(b) Mahdieh encrypted image

(c) Kekre encrypted image,

(d) proposed encrypted image

(e) Reconstructed image of non-encrypted

(f) Reconstructed image of Mahdieh

(g) Reconstructed image of Kekre

(h) Reconstructed image of proposed

Figure 9. Reconstructed lena image of JPEG2000 (CR=10)

The JPEG200 lossless compression's BPP result is indicated in Table 3. The Mahdieh [15], Wang [6], Zhang [7] and Wang [8] algorithms require more than 8.22 bit requires to lossless compression the test images. The non-compressed image with 1024×1024, grayscale requires 8 bpp, the average value is higher than non-compressed bpp. Also, the average bpp of Kekre's work is 4.85 bpp more than 2.01 bpp compared to non-encrypted image. However, the average value of bpp of proposed works is 2.98 bpp and more than 0.14 bpp compared to non-encrypted image. In the case of Chest01, Chest02 and Ankle, the proposed work requires smaller bits compared to non-encrypted. These results indicate that the proposed encryption algorithm can works with lossless compression algorithm. In Figure 8(a) to Figure 8(e) shows the PSNR measurement of JPEG and Figure 8(g) to Figure 8(l) the SSIM measurement of JPEG comparison between previous, non-encrypted and proposed work with different bpp. The result shows that PSNR and SSIM value of our proposed work are greater than other encryption works and non-encrypted images with high compression ratio.

The proposed algorithm provides an encryption algorithm that is convenient with both lossy and lossless compression algorithms. The values of PSNR and SSIM in the results show that compression efficiency improves, and the other analysis shows that the proposed algorithm provides security.

Table 3. JPEG2000 lossless compression's BPP of previous and proposed algorithms

Test Image	Non Encrypted	Mahdieh [15]	Kekre [22]	Wang [6]	Zhang [7]	Wang [8]	Proposed work
1. Shoulder	3.22	8.38	5.46	8.70	8.38	8.66	3.59
2. Chest00	2.72	8.68	5.19	8.71	8.09	8.68	3.13
3. Chest01	3.06	8.53	4.83	8.72	8.26	8.68	2.95
4. Chest02	2.90	8.61	4.68	8.70	8.20	8.69	2.81
5. Ankle	2.35	8.87	3.24	8.73	7.93	8.67	2.22
6. Lena	2.76	8.05	5.71	8.71	8.43	8.68	3.15
Average	2.84	8.52	4.85	8.71	8.22	8.68	2.98

4 Conclusion

In this paper, we described a compression-friendly image encryption algorithm based on the order relation theory. In the process, horizontal and vertical permutations remove the correlation between neighboring subsets to achieve encryption. Each directional permutation method has two steps, application of the order relation to the horizontal and vertical data, then permuting the subsets. The security of our proposed method was confirmed via correlation analysis, mean square analysis and key space analysis. The key space analysis has ensured the security of proposed work, and indicates that generated encryption key was related with the original image size. For future work, we will reduce the encryption key of encrypted image. The results of the compression-friendly analysis show that the proposed encryption method with JPEG and JPEG200 compression, the PSNR and SSIM values are significantly higher than those when using existing methods and non-encrypted images. The proposed encryption is independent of the compression algorithm being capable to employ with any existing compression standard.

References

[1] H. Cheng, Y. Song, C. Huang, Q. Ding, Self-Adaptive Chaotic Logistic Map: An Efficient Image Encryption Method, *Journal of Internet Technology*, Vol. 17, No. 4, pp. 743-752, July, 2016.

[2] G. Gankhuyag, S. Hong, Y. Choe, Compression Friendly Medical Image Encryption Based Order Relation, *2013 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2013, pp. 568-569.

[3] W. Diffie, M. E. Hellman, New Directions in Cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, November, 1976.

[4] M. Liu, F. Zhao, X. Jiang, X. Liu, Y. Liu, A Novel Image Encryption Algorithm Based on Plaintext-related Hybrid Modulation Map, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2141-2155, December, 2019.

[5] S. Rakesh, A. A. Kaller, B. C. Shadakshari, B. Annappa, Image Encryption Using Block Based Uniform Scrambling and Chaotic Logistic Mapping, *International Journal on Cryptography and Information Security*, Vol. 2, No. 1, pp. 49-57, March, 2012.

[6] X. Y. Wang, L. Yang, R. Liu, A. Kadir, A Chaotic Image Encryption Algorithm Based on Perceptron Model, *Nonlinear Dynamics*, Vol. 62, No. 3, pp. 615-621, November, 2010.

[7] Y. Q. Zhang, X. Y. Wang, A New Image Encryption Algorithm Based on Non-adjacent Coupled Map Lattices, *Applied Soft Computing*, Vol. 26, pp. 10-20, January, 2015.

[8] X. Wang, L. Liu, Y. Zhang, A Novel Chaotic Block Image Encryption Algorithm Based on Dynamic Random Growth Technique, *Optics and Lasers in Engineering*, Vol. 66, pp. 10-18, March, 2015.

[9] H. Liu, X. Wang, Color Image Encryption Based on One-time Keys and Robust Chaotic Maps, *Computers & Mathematics with Applications*, Vol. 59, No. 10, pp. 3320-3327, May, 2010.

[10] H. Liu, X. Wang, Color Image Encryption Using Spatial Bit-level Permutation and High-dimension Chaotic System, *Optics Communications*, Vol. 284, No. 16-17, pp. 3895-3903, August, 2011.

[11] H. Liu, X. Wang, A. Kadir, Image Encryption Using DNA Complementary Rule and Chaotic Maps, *Applied Soft Computing*, Vol. 12, No. 5, pp. 1457-1466, May, 2012.

[12] X. Y. Wang, Y. Q. Zhang, X. M. Bao, A Novel Chaotic Image Encryption Scheme Using DNA Sequence Operations, *Optics and Lasers in Engineering*, Vol. 73, pp. 53-61, October, 2015.

[13] A. Mitra, Y. V. S. Rao, S. R. M. Prasanna, A New Image Encryption Approach Using Combinational Permutation Techniques, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.360.989&rep=rep1&type=pdf

[14] S. Manimurugan, K. Porkumaran, Secure Medical Image Compression Using Block Pixel Sort Algorithm, *European Journal of Scientific Research*, Vol. 56, No. 2, pp. 129-138, July, 2011.

[15] M. K. Lakhani, H. Behnam, A. Karimi, Secure Transmission of Images Based on Chaotic Systems and Cipher Block Chaining, *Journal of Electronic Imaging*, Vol. 22, No. 1, Article ID 013025, February, 2013.

[16] Y. Piao, D. Shin, E. Kim, Robust Image Encryption by Combined Use of Integral Imaging and Pixel Scrambling Techniques, *Optics and Lasers in Engineering*, Vol. 47, No. 11, pp. 1273-1281, November, 2009.

[17] Z. Parvin, H. Seyedarabi, M. Shamsi, A New Secure and Sensitive Image Encryption Scheme Based on New Substitution with Chaotic Function, *Multimedia Tools and Applications*, Vol. 75, No. 17, pp. 10631-10648, September, 2016.

[18] C. Zhu, A Novel Image Encryption Scheme Based on Improved Hyperchaotic Sequences, *Optics Communications*, Vol. 285, No. 1, pp. 29-37, January, 2012.

[19] J. Zhao, H. Lu, X. Song, J. Li, Y. Ma, Optical Image Encryption Based on Multistage Fractional Fourier Transforms and Pixel Scrambling Technique, *Optics Communications*, Vol. 249, No. 4, pp. 493-499, May, 2005.

[20] R. Ye, A Novel Chaos-based Image Encryption Scheme with an Efficient Permutation-diffusion Mechanism, *Optics Communications*, Vol. 284, No. 22, pp. 5290-5298, October, 2011.

[21] Z. H. Guan, F. Huang, W. Guan, Chaos-based Image Encryption Algorithm, *Physics Letters A*, Vol. 346, No. 1, pp. 153-157, October, 2005.

[22] H. B. Kekre, T. Sarode, P. N. Halarnkar, Study of Perfect Shuffle for Image Scrambling, *International Journal of Scientific and Research Publications*, Vol. 4, No. 2, pp. 1-7, February, 2014.

[23] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J-J. Quisquater, Overview on Selective Encryption of Image and Video: Challenges and Perspectives, *EURASIP Journal on Information Security*, Vol. 2008, Article number 179290, December, 2008.

[24] L. Zhang, Z. Zhu, B. Yang, W. Liu, H. Zhu, M. Zou, Medical Image Encryption and Compression Scheme Using Compressive Sensing and Pixel Swapping Based Permutation Approach, *Mathematical Problems in Engineering*, Vol. 2015, Article ID 940638, August, 2015.

[25] N. Zhou, A. Zhang, J. Wu, D. Pei, Y. Yang, Novel Hybrid Image Compression-encryption Algorithm Based on Compressive Sensing, *Optik*, Vol. 125, No. 18, pp. 5075-5080, September, 2014.

[26] S. Lipschutz, *Schaum's Outline of Theory and Problems of Set Theory and Related Topics*, McGraw Hill, 1995

[27] P. R. Halmos, *Naive Set Theory*, Springer, 1960.

[28] E. Schechter, *Handbook of Analysis and Its Foundations*, Academic Press, 1996.

[29] M. J. Sabin, R. M. Gray, Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm, *IEEE Transactions on Information Theory*, Vol. 32, No. 2, pp. 148-155, March, 1986.

[30] R. M. Rad, A. Attar, R. E. Atani, A Comprehensive Layer Based Encryption Method for Visual Data, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 6, No. 1, pp. 37-48, February, 2013.

[31] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, M. R. Mosavi, A Novel Image Encryption Based on Hash Function with Only Two-round Diffusion Process, *Multimedia Systems*, Vol. 20, No. 1, pp. 45-64, February, 2014.

[32] C. Y. Song, Y. L. Qiao, X. Z. Zhang, An Image Encryption Scheme Based on New Spatiotemporal Chaos, *Optik - International Journal for Light and Electron Optics*, Vol. 124, No. 18, pp. 3329–3334, September, 2013.

[33] C. Christopoulos, A. Skodras, T. Ebrahimi, The JPEG2000 Still Image Coding System: An Overview, *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, pp. 1103-1127, November, 2000.

Biographies

Ganzorig Gankhuyag received a B.S. in Information and Communication Engineering from HUREE ICT University, Ulaanbaatar, Mongolia, in 2006, and an M.S. degree in Electronic Engineering from Konkuk University, Seoul, Korea, in 2009. From 2009 to 2011, he was a lecturer at HUREE ICT University, Ulaanbaatar, Mongolia. He is currently pursuing a Ph.D. degree at Yonsei University, Seoul, Korea. His research interests include image and video compression, SoC design, embedded processes, and cryptography.

Yoonsik Choe received a B.S. in Electrical Engineering from Yonsei University, Seoul, Korea in 1979. He also received an M.S.E.E in Systems Engineering at Case Western Reserve University, Cleveland, OH, in 1984, an M.S. in Electrical Engineering from Pennsylvania State University, University Park, PA, in 1987, and a Ph.D. in Electrical Engineering from Purdue University, West Lafayette, IN, in 1990. From 1990 to 1993, he was a Principal Engineer at Hyundai Electronics. Since 1993, he has been with the Department of Electrical and Electronic Engineering at Yonsei University, Seoul Korea, where he is a Professor. His research interests include video coding, video communication, statistical signal processing, and digital image processing.