Soft Computing (2020) 24:3829–3848
https://doi.org/10.1007/s00500-019-04151-8

METHODOLOGIES AND APPLICATION

A new image encryption algorithm using random numbers generation of two matrices and bit-shift operators

Mohammed Es-Sabry[1] · Nabil El Akkad[1,2] · Mostafa Merras[1,3] · Abderrahim Saaidi[1,4] · Khalid Satori[1]

Published online: 25 June 2019

Abstract
In this work, we proposed a new approach to encrypt color images using two matrices with size of 16 × 16 whose integer values are between 0 and 255 generated randomly, and the bit-shift operators. These matrices are used to perform the first encryption phase. The first value of the first matrix is calculated from the pixels of each channel (red, green and blue) of the original image; the rest of the values are randomly generated; each value must be unique; the values of the second matrix are unique and generated randomly. The first encryption phase of the original image is done by digraph (two-digit sequence). We take the first digit in the first matrix, the second digit in the second matrix; then, we look in these matrices for the numbers that complete the rectangle. In the second encryption phase, we used a right circular shift of bits; the number of bits to shift is calculated according to a function which considers the values of the two matrices as well as their positions (row and column). Therefore, any change in the two keys (two matrices) will completely change the encrypted image. Our encryption system is resistant against brute force attacks, statistical attacks as well as differential attacks. The results are justified by applying several safety criteria, such as correlation coefficient, entropy and peak signal-to-noise ratio (PSNR). In addition, our method is very sensitive to the change made, either in the original image or in the two keys used for the encryption, which was justified by calculating the number of changing pixel rate (NPCR > 99.69) and the unified averaged changed intensity (UACI > 33.54).

Keywords Image encryption · Security · Random numbers generation · Bit-shifting operators

1 Introduction

In recent years, digital networks have evolved significantly and have become unavoidable for modern communication. The images transmitted on these networks are particular data because of their large amount of information. As a result, a lot of research has been done on these images to prevent access by illegal users and attackers. The purpose of image encryption (Es-Sabry et al. 2018a, b) is to convert the original image into another image that cannot be interpreted; it is essential that no one is able to decipher the image without the appropriate decryption key. It has been found that traditional encryption schemes are not suitable for image encryption due to the strong correlation between the pixels in the image as well as the data capacity which is huge; this is why algorithms like AES and DES

Communicated by V. Loia.

✉ Mohammed Es-Sabry
mohammed.es.sabry@usmba.ac.ma

Nabil El Akkad
nabil.elakkad@usmba.ac.ma

Mostafa Merras
merras.mostafa@gmail.com

Abderrahim Saaidi
abderrahim.saaidi@usmba.ac.ma

Khalid Satori
khalidsatori@gmail.com

[1] LIIAN, Department of Mathematics and computer science, Faculty of Sciences, Dhar-Mahraz, Sidi Mohamed Ben Abdellah University, B.P 1796 Atlas, Fez, Morocco

[2] Department of Electrical and Computer Engineering, National School of Applied Sciences (ENSA), Sidi Mohamed Ben Abdellah University, Fez, Morocco

[3] Department of Computer Science, High School of Technology, Moulay Ismail University, Meknes, Morocco

[4] LSI, Department of Mathematics, physics and Informatics, Polydisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza, Morocco

are no longer used in this field, since the key space is not large enough which seriously affects the confidentiality of the information network, so they take a lot of time in computing phase.

Due to the good properties of chaotic encryption, such as sensitivity to initial conditions and system parameters, pseudo-random property, non-periodicity and topological transitivity, many chaos-based image encryption methods have recently been proposed and have been widely used in several fields such as biology (Lesne 2006), electrical engineering (Liu and Lu 2010), physics (Weidenmuller and Mitchell 2009) and complex networks (Lu et al. 2012), not to mention its easy implementation and speed, which makes it very useful for the communication of images in real time. Chaotic maps can be categorized into one-dimensional (1D) chaotic maps and high-dimensional (HD) chaotic maps. 1D chaotic maps usually contain one variable and some parameters, such as, the Gaussian, Logistic, Sinus and Tent maps. These maps are characterized by the simplicity of their structures and their chaotic orbits. With the development of chaotic signal estimation technologies, when little information is extracted, their chaotic orbits can be estimated (Arroyo et al. 2008; Ling et al. 1999) and their parameters and initial values can be predicted (Wu et al. 2004). These weaknesses limit their applications in many fields of security. For example, when 1D chaotic maps are used in image encryption, several encryption algorithms have been reported as unsecured (Arroyo et al. 2008; Skrobek 2007).

Compared to 1D chaotic maps, HD chaotic maps usually have more complex structures and better chaotic performance. These make their chaotic orbits much more difficult to predict. However, their hardware implementations are relatively complex and expensive. Therefore, developing a chaotic map with excellent chaotic performance and a low implementation cost becomes significative.

On the other hand, there is another encryption system based on the random generation of numbers such as:

- True random number generators (TRNGs).
- Pseudo-random number generators (PRNGs).
- Chaotic random number generators (CRNGs).

The first category (TRNGs) is based on the use of unpredictable and non-deterministic sources such as natural processes or physical phenomena, so it is impossible to predict the next number, knowing that the previous numbers are known. For this reason, TRNGs are particularly useful in cryptography and especially in the production of keys. The second category (PRNGs) is the algorithmic generators of numbers that have the appearance of random but with periodic digit sequences, nevertheless, their results are predictable. Good random number generators produce very long sequences that seem random, in the sense that no efficient algorithm can guess the next number according to the prefix of the sequence. The third category (CRNGs) is based on the chaotic phenomenon; it is a multiparametric system; these parameters must be refined to improve its performances, although they are periodic but by applying spatiotemporal techniques, their period can last several years, so practically inaccessible.

The authors, in Zhang et al. 2005, use a discrete exponential chaotic map in order to permute the images pixels; after that, they use the exclusive OR (XOR) operation in time domain. In Li et al. 2009, a new image encryption scheme was proposed using two chaotic logistic maps which were used with an external secret key of 80 bits. The scheme updates the secret key after encrypting each block of image pixels. The authors suggested in Fridrich 1997, a chaotic image encryption method in time domain. All the pixels are moved using a 2D chaotic map and their values are altered sequentially. In Hwang et al. 2007, the authors proposed an image copyright protection scheme using the computational integral imaging (CII) technique, in which the 3D watermark information is embedded into the DWT domain. The 3D watermark plain image can be reconstructed using a CIIR technique. This scheme is highly robust because of the data redundancy of the elemental images. However, in the watermark extraction process, CIIR is a pixel superposition reconstruction method. The resolution of the reconstructed 3D plain image is dramatically degraded as the magnification factor increases. In Wang et al. 2006, the authors proposed a fully digital encryption system based on extended Fractal Fourier Transform (FRT) and digital holography technique. The parameters of the extended FRT gave an enhancement on security compared with the conventional FRT. The input image was extended fractional Fourier transformed two times and encrypted by use of a random phase mask in the extended FRT encryption system. By use of an interference with a wave from another phase mask, the data were encrypted again and recorded as a digital hologram. The decryption key could also be recorded as the key hologram as described in the following section. The parameters, the random phase code and the key hologram form the keys to the encrypted image. The authors developed in Piao et al. 2009, an image encryption scheme based on integral imaging and pixel scrambling techniques. In the scheme, the input image is first recorded using the CII technique and the obtained elemental images are scrambled by a pixel scrambling technique. In the reconstruction process, the image is reconstructed by the CIIR technique. This method is a highly secure and robust encoding system. However, it incurs resolution degradation problems. The authors in Singh and Sinha 2008 proposed a method for image encryption using Fractal Fourier Transform (FRT) and chaos. The image to be encrypted was FRT two times, and

the random phase masks generated using chaos functions were placed in the intermediate planes. The chaos functions had been used to generate the chaotic random phase masks (CRPM). The comparison of the mean square errors (MSE) and signal-to-noise ratio (SNR) between the decrypted image and the input image for correct order of FRT and incorrect order of FRT using these chaos functions had been done.

The rest of this work is organized as follows: The second part presents the proposed method. The experimental results and the analysis of the performance of the method are covered in the third part. In the last part, a conclusion of this work is presented.

1.1 Proposed method

Our method is based on use of two 16 × 16 matrices which will be used to encrypt the original image. Each matrix is composed of different values between 0 and 255 and is generated randomly except for the first value of the first matrix which is calculated from the three channels (red, green and blue) of the original image (Fig. 1). Figure 2 represent a flowchart that shows the different steps to encrypt a color image (Fig. 3).

The encryption operation of the color image requires to encrypt each image extracted from the three channels and then merge them, which will give us an encrypted color image at the output.

Given a color image to be encrypted, the first step is to decompose the color image into three channels, red channel, green channel and blue channel; then, we will extract the decimal value matrix for each channel (Fig. 3); these values are between 0 and 255; for each channel, we will calculate the sum of all pixels modulo 255; then, we will use the floor function to extract the greatest integer less than or equal the average of the values obtained; the result will be an integer between 0 and 255; after that, we will put this value in the first cell of the first matrix, the rest of the cells of the matrix will be filled by different values generated randomly between 0 and 255. The same procedure will be applied for all the cells of the second matrix.

The calculation of the first cell of the first matrix according to the pixels of the original image will be justified in the experimentation part.

Figure 3 shows an example of 16 pixels matrix extracted from a 256 × 256 color image after decomposing it into three channels.

After performing the generation operation of the two matrices, decomposing the color image into three channels and extracting the decimal values matrix for each channel, we proceed to the encryption operation.

The values of each matrix are encrypted by digraph (two-digit sequence). For example, if we want to encrypt

Fig. 1 16 × 16 matrix containing values between 0 and 255

164	250	132	167	189	48	155	98	67	187	45	166	21	102	137	242
204	168	34	160	74	27	141	226	251	205	52	134	181	10	39	38
13	90	0	246	207	153	19	196	95	245	97	59	9	175	8	147
161	143	7	216	228	213	22	229	41	188	223	218	25	99	128	240
88	198	55	28	113	194	210	192	254	30	80	138	174	248	16	235
215	117	125	191	154	197	146	4	60	91	114	159	201	96	3	220
185	150	241	209	1	170	173	208	75	24	139	255	156	32	202	57
186	214	15	123	212	162	227	6	244	222	206	93	112	109	176	61
238	44	165	121	58	172	211	183	221	82	20	234	76	108	89	130
237	145	104	69	133	46	105	43	182	180	37	177	136	110	33	135
12	119	47	120	65	118	230	149	152	71	35	5	190	171	92	62
184	232	54	106	63	68	224	14	203	115	87	11	225	42	17	122
40	101	79	126	157	86	36	247	73	100	151	179	142	124	163	26
148	77	64	85	140	169	193	231	158	253	23	199	195	29	107	66
127	51	131	200	78	103	18	84	56	83	50	219	252	243	53	178
239	111	94	144	70	217	249	72	49	129	233	31	116	236	2	81

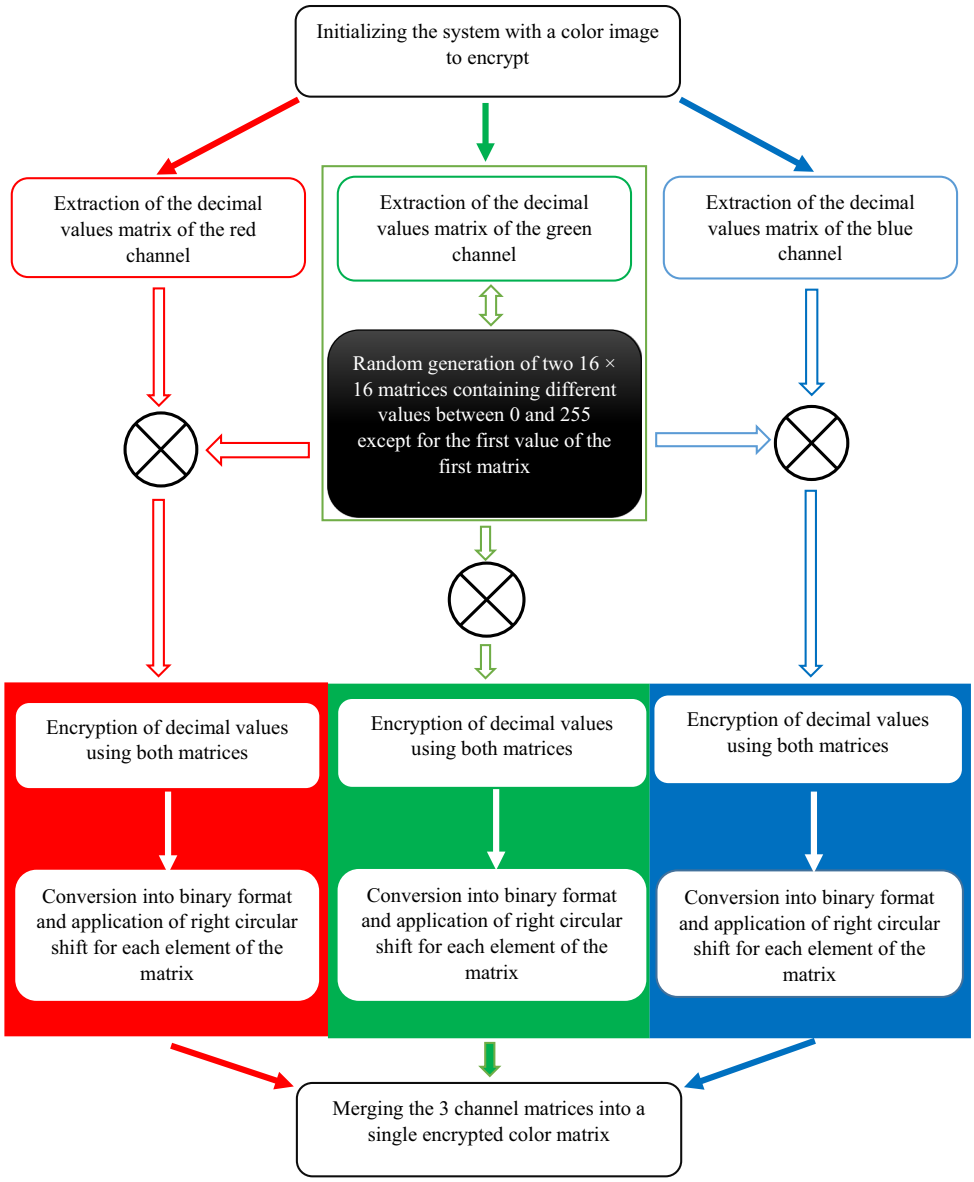


Fig. 2 Proposed method flowchart

the digraph 202,58. We look for the number 202 in the matrix 1, the number 58 in the matrix 2, and then we look in these matrices for the numbers that complete the rectangle; in our example, Fig. 4, the number 209 located in the matrix 1 and the number 199 located in the matrix 2. The digraph 202,58 is encrypted in digraph 199,209, because by convention, one of the two encrypted digits is on the same column as the digit of the original matrix. If both digits are in the same column, their inversion form will be the encrypted digraph. For example, 75,184 becomes 184,75 (Fig. 5).

The problem is that this method is vulnerable to cryptanalytic attacks; each digraph has only one form of encryption (Fig. 6), which will greatly reduce the number of tests of a cryptanalytic brute force attack based on frequency analysis of digraphs.

In the example, we applied our method on the 16 pixels extracted from the red channel of the example (Fig. 3); if we analyze the results obtained, we note that the digraph 202,58 is repeated three times and digraph 1,12 repeats itself twice. So, with a cryptanalytic brute force attack based on the frequency analysis of the digraphs, we will obtain the following results (Table 1).

The results obtained in the table are calculated as follows:

We have a 4 × 4 size matrix that contains integers between 0 and 255; so the matrix contains 16 cells; each cell can take 256 values, and therefore a brute force attack will test 256^{16} possibilities at worst case to find the original matrix; however, if we take into account the repeated

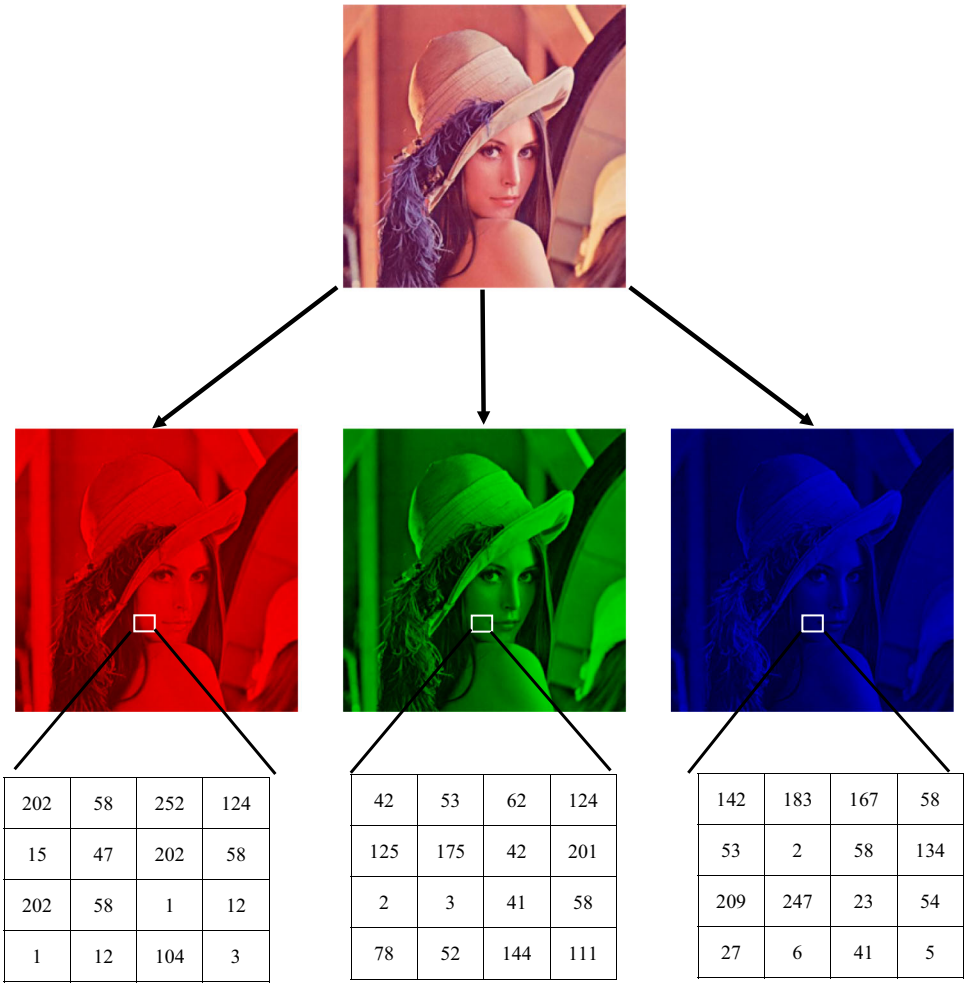


Fig. 3 Decomposition of the color image on three channels and extraction of the decimal matrices

digraphs, we will eliminate them since they will take the same value as that of the first occurrence, so an attack of brute force cryptanalytic will test 256^{10} possibilities; these results show that there is a large difference in the number of possibilities between an attack of the brute force and a brute force cryptanalytic attack based on the frequency analysis of the digraphs. So, to make our method invulnerable for this type of attack, we will add a second phase of encryption based on the use of right circular shift of the binary matrix of the decimal values matrix obtained from the first phase of encryption; these values are encoded on 8 bits. The number of bits needed to make the circular shift must follow the following formula:

$$N(X) = (K_1 + K_2 + i(X) + j(X)) mod\, 8 \quad (1)$$

With N is a function that calculates the number of bits needed to make the rotation, X represents a binary value, i and j are two functions that return, respectively, the row and column index of the X value, K_1 and K_2 are calculated from the 16×16 encryption matrices, and they are extremely important since any changes made at the keys (two matrices) will completely change the encrypted image. These values are calculated from the following formulas.

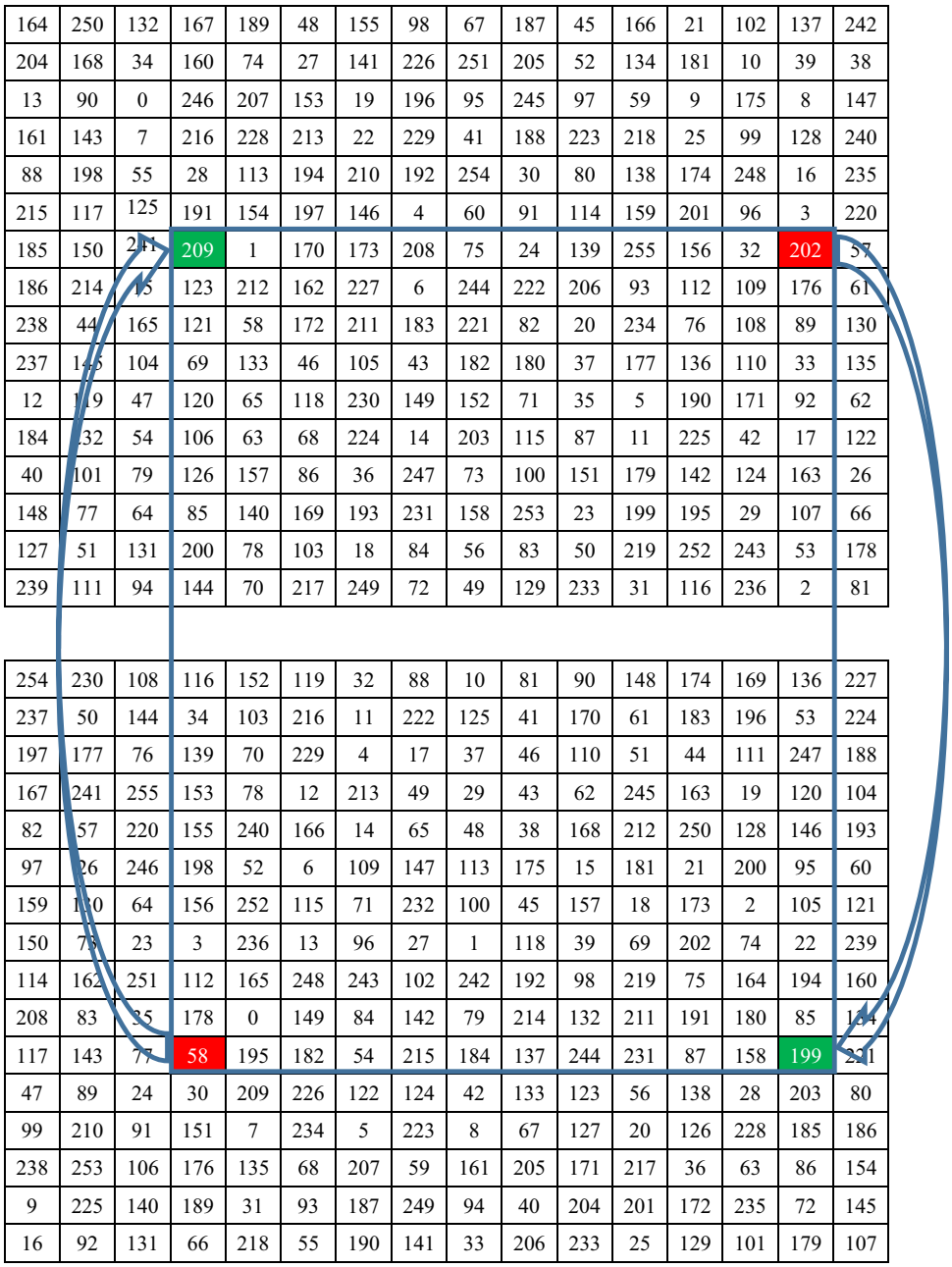
$$K_1 = \left(\sum_{i=0}^{15} \sum_{j=0}^{15} M_{ij}^1 \times (i+j) \right) mod\ 8 \quad (2)$$

$$K_2 = \left(\sum_{i=0}^{15} \sum_{j=0}^{15} M_{ij}^2 \times (i+j) \right) mod\ 8 \quad (3)$$

With M_{ij}^1, the value of the ith line and the jth column of the first matrix, and M_{ij}^2 the value of the ith line and the jth column of the second matrix.

In Fig. 7, we applied the second encryption phase based on the right circular shift of the bits with $K_1 = 1$ and $K_2 = 6$; these values are obtained by applying Eqs. (2) and (3) on the matrices of Fig. 4. We note that the digraph **199,209** which was repeated three times before this phase has no longer the same encrypted form at the output and the same for the digraph **78,170**; therefore, any cryptanalytic attack based on the frequency analysis of the digraphs will no longer have any influence on the image decryption.

Fig. 4 Encryption operation of the digraph **202,58** located in different columns



164	250	132	167	189	48	155	98	67	187	45	166	21	102	137	242
204	168	34	160	74	27	141	226	251	205	52	134	181	10	39	38
13	90	0	246	207	153	19	196	95	245	97	59	9	175	8	147
161	143	7	216	228	213	22	229	41	188	223	218	25	99	128	240
88	198	55	28	113	194	210	192	254	30	80	138	174	248	16	235
215	117	125	191	154	197	146	4	60	91	114	159	201	96	3	220
185	150	241	209	1	170	173	208	75	24	139	255	156	32	202	57
186	214	15	123	212	162	227	6	244	222	206	93	112	109	176	61
238	44	165	121	58	172	211	183	221	82	20	234	76	108	89	130
237	145	104	69	133	46	105	43	182	180	37	177	136	110	33	135
12	119	47	120	65	118	230	149	152	71	35	5	190	171	92	62
184	232	54	106	63	68	224	14	203	115	87	11	225	42	17	122
40	101	79	126	157	86	36	247	73	100	151	179	142	124	163	26
148	77	64	85	140	169	193	231	158	253	23	199	195	29	107	66
127	51	131	200	78	103	18	84	56	83	50	219	252	243	53	178
239	111	94	144	70	217	249	72	49	129	233	31	116	236	2	81

254	230	108	116	152	119	32	88	10	81	90	148	174	169	136	227
237	50	144	34	103	216	11	222	125	41	170	61	183	196	53	224
197	177	76	139	70	229	4	17	37	46	110	51	44	111	247	188
167	241	255	153	78	12	213	49	29	43	62	245	163	19	120	104
82	157	220	155	240	166	14	65	48	38	168	212	250	128	146	193
97	26	246	198	52	6	109	147	113	175	15	181	21	200	95	60
159	130	64	156	252	115	71	232	100	45	157	18	173	2	105	121
150	75	23	3	236	13	96	27	1	118	39	69	202	74	22	239
114	162	251	112	165	248	243	102	242	192	98	219	75	164	194	160
208	83	35	178	0	149	84	142	79	214	132	211	191	180	85	134
117	143	77	58	195	182	54	215	184	137	244	231	87	158	199	221
47	89	24	30	209	226	122	124	42	133	123	56	138	28	203	80
99	210	91	151	7	234	5	223	8	67	127	20	126	228	185	186
238	253	106	176	135	68	207	59	161	205	171	217	36	63	86	154
9	225	140	189	31	93	187	249	94	40	204	201	172	235	72	145
16	92	131	66	218	55	190	141	33	206	233	25	129	101	179	107

2 Experimental results and performance analysis

To say that an encryption system is good, it must resist most attacks that exist in the field such as statistical attacks, differential attacks as well as different attacks of brute force; that is why, we have measured the strength of our proposed method and its resistance against the various attacks mentioned above.

To do this, we took eight color images with different sizes (256×256, 512×512 and 1024×1024); we applied our own encryption system on these images using java programming language.

The performance of our proposed algorithm is justified based on several criteria which will be explained at length in the next sections as well as the comparison with other methods.

2.1 Histogram analysis

The first criterion consists in analyzing histograms of the original color images and those images encrypted with our

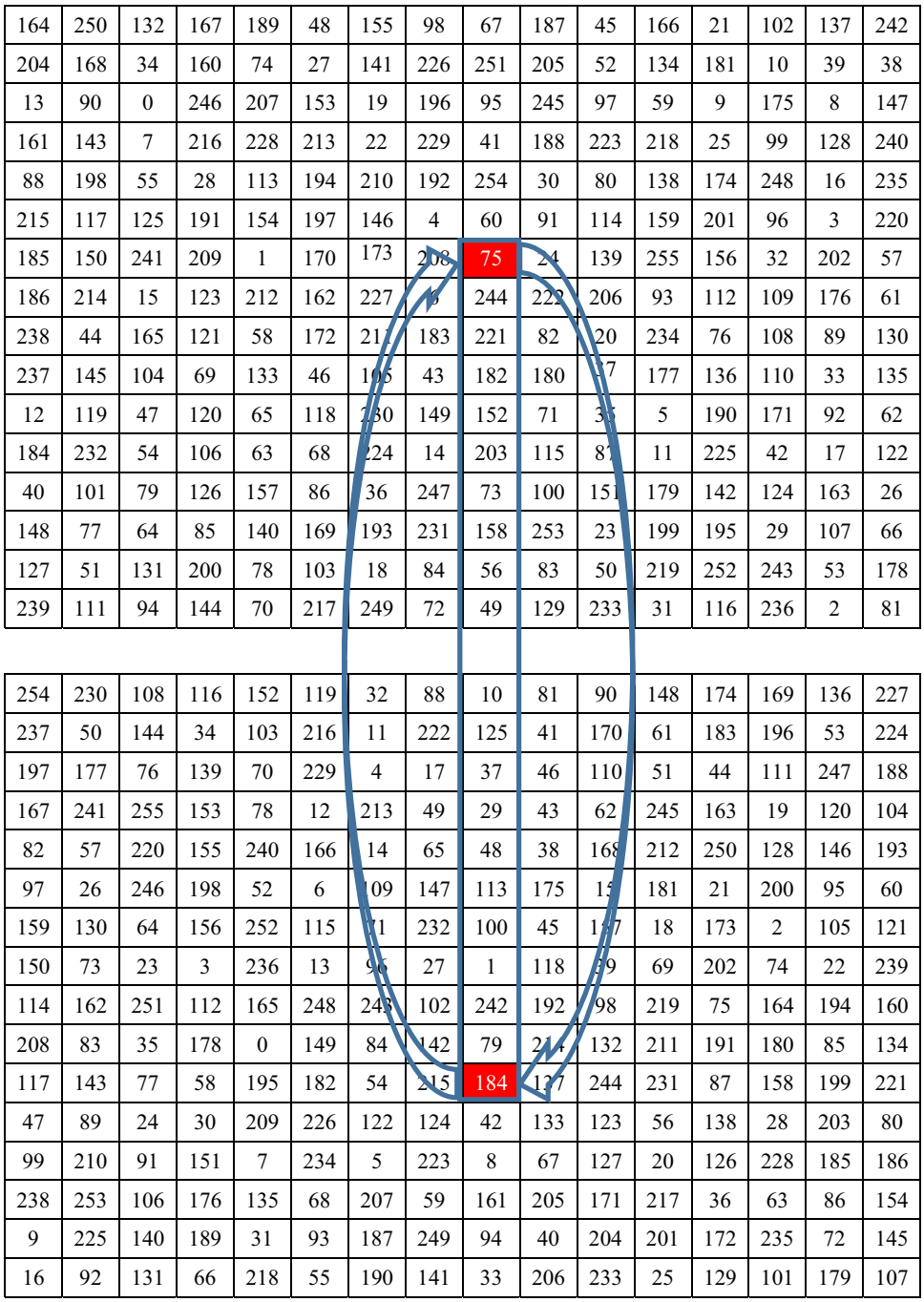


Fig. 5 Encryption operation of the digraph 75,184 located in the same columns

own algorithm and make a comparison between them in terms of the distribution of the pixels for each channel (red, green and blue).

From Figs. 8, 9, 10, 11, 12, 13, 14, 15 and 16 it is noted that not only the encrypted images histograms are completely different from those of the original images, but also the pixels distribution is almost uniform; hence, our method can resist any statistical attacks and finally increases the protection of the information included in the images.

2.2 Correlation analysis of two adjacent pixels

The correlation coefficient of adjacent pixels is an important internal information of the image; the statistical attacks use this kind of information to decrypt images; therefore, a good encryption system must get rid of this information by minimizing the correlation coefficient as much as possible.

To do this, we took 5000 pairs of randomly selected adjacent pixels from the original image and the encrypted image along three directions (horizontal, vertical and

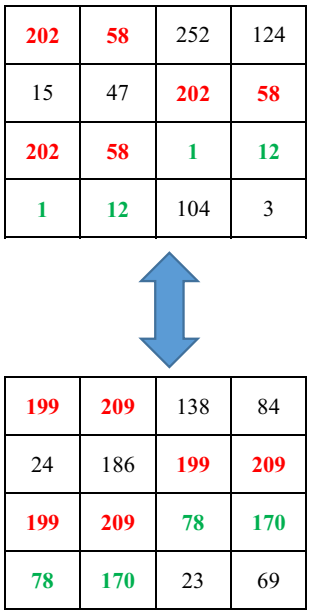


Fig. 6 Encryption operation of a 4 × 4 matrix

Table 1 Results obtained for each type of attack

Brute force normal	$256^{16} \approx 3.2 \times 10^{38}$
Brute force cryptanalytic	$256^{10} \approx 1.6 \times 10^{24}$

diagonal) for each of the three channels (red, green and blue).

In this test, the calculation is applied to the images of Figs. 8, 9, 10 and 11 using the following equations:

$$r_{xy} = \frac{Cov(x,y)}{\sqrt{D_x}\sqrt{D_y}} \tag{4}$$

$$Cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E_x)(y_i - E_y) \tag{5}$$

$$D_x = \frac{1}{N}\sum_{i=1}^{N}(x_i - E_x)^2 \tag{6}$$

$$E_x = \frac{1}{N}\sum_{i=1}^{N}x_i \tag{7}$$

With N represents the number of pixels of the original image.

The table below shows the results obtained after calculating the correlation coefficient; Figs. 17, 18 and 19 show the correlation distribution of the pairs of adjacent pixels in the three channels (red, green and blue) and following the three directions (horizontal, vertical and diagonal).

We note that the correlation coefficient (CC) in the three channels (red, green and blue) including the three directions (horizontal, vertical and diagonal) takes values close to 1 in the case of original images. On the other hand, it takes values close to 0 in the case of encrypted images, which means that the degree of dependence between the two adjacent pixels is very strong before the application of our method, and this dependence is destroyed after having applied our encryption method (Table 2).

Figures 17, 18 and 19 show the distribution of pixels in three channels (red, green and blue) and following three directions (horizontal, vertical and diagonal) of the Lena image. We can see that the distribution before the application of our method follows a line, which shows that the correlation is strong between the pixels. On the other hand, this correlation becomes weak after applying our approach, which means that the proposed method gives good results. The figures show that the pixels are scattered everywhere.

Table 3 represents the correlation coefficients (CC) obtained by our approach compared with method of Tong et al. (2009), method of Borujeni and Eshghi (2011), method of Wang and Guo (2014), method of Hua (Hua et al. 2015) and method of Tong et al. (2015).

Since we worked on three channels (red, green and blue), we took the average for each channel following each direction (horizontal, vertical and diagonal); these values are calculated by applying our method on Lena image (Fig. 8).

The results show that the correlation existing between the adjacent pixels has been destroyed by our algorithm, which shows the performance of our approach compared to other methods.

2.3 Correlations between original and encrypted images

After having made the visual test by analyzing the histograms of original images and encrypted images for each channel (red, green and blue) and after calculating the correlation coefficient of two adjacent pixels, we will now calculate the correlation coefficient (CC) for each channel of the original image with that of the encrypted image by applying the following formula:

$$CC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(A_{ij} - \bar{A})^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(B_{ij} - \bar{B})^2}} \tag{8}$$

$$\bar{A} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}A_{ij} \tag{9}$$

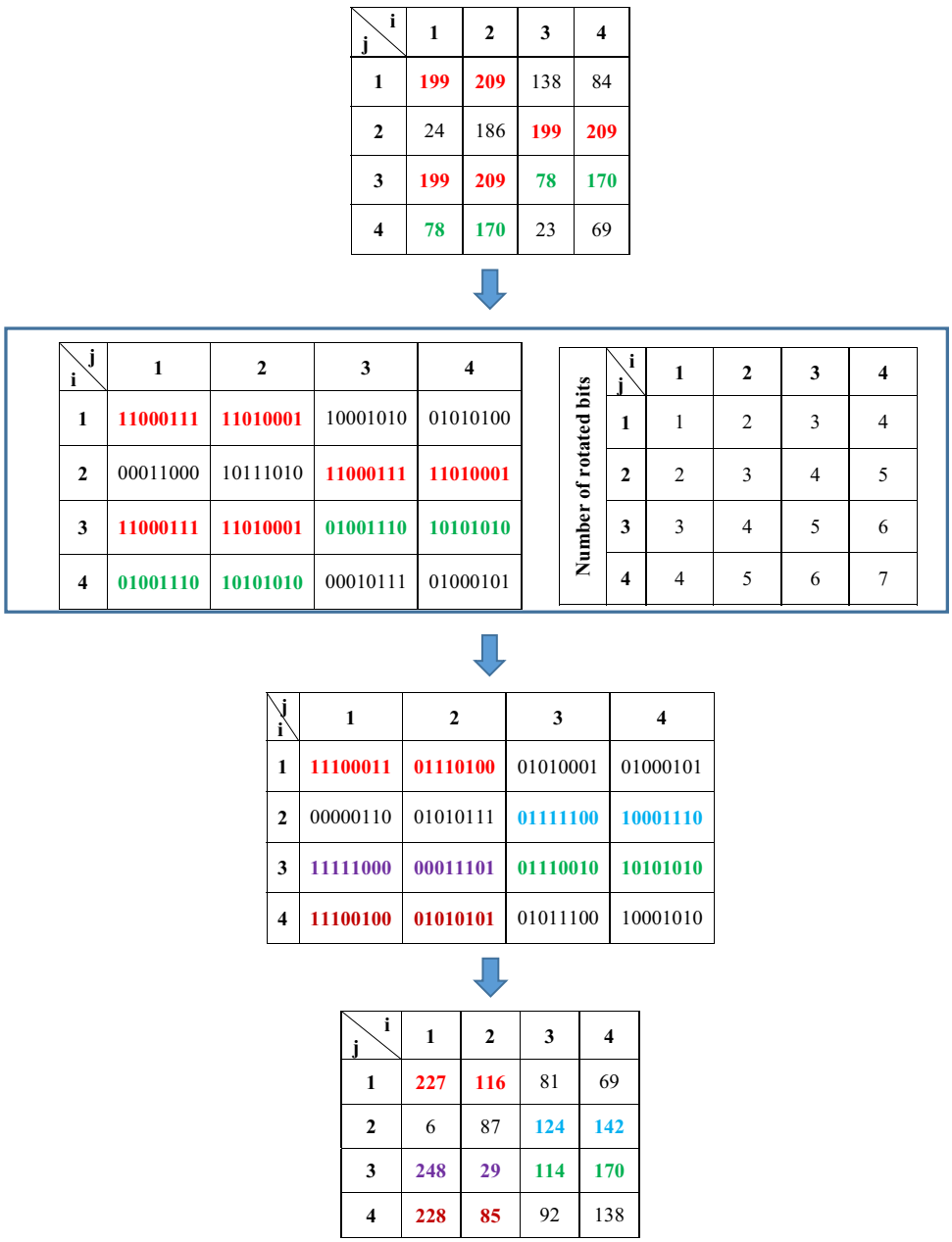


Fig. 7 Right circular shift of a 4×4 matrix

$$\bar{B} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} B_{ij} \tag{10}$$

Where A represents the original image, B represents the encrypted image, $\bar{A}$ and $\bar{B}$ represent, respectively, the mean values of the elements of matrices A and B, M and N represent, respectively, the height and the width of matrices A and B.

Table 4 shows the different values obtained after calculating the correlation coefficient (CC) for each channel (red, green and blue) using Figs. 8, 9, 10, 11, 12, 13, 14 and 15; the fifth column contains the average value of the three channels for each image; this value is compared later with the value obtained by the Zhu method (Zhu 2012).

The results show the performance of our method in terms of the correlation between the original image and the encrypted image since the values are very close to 0; these values are better than those obtained by the Zhu method.

2.4 Resistance to differential attack

Sometimes the attackers just make a small change at a pixel of the original image to see the effect of this change on the encrypted image; a good encryption system must take into account these changes; after the change made, the

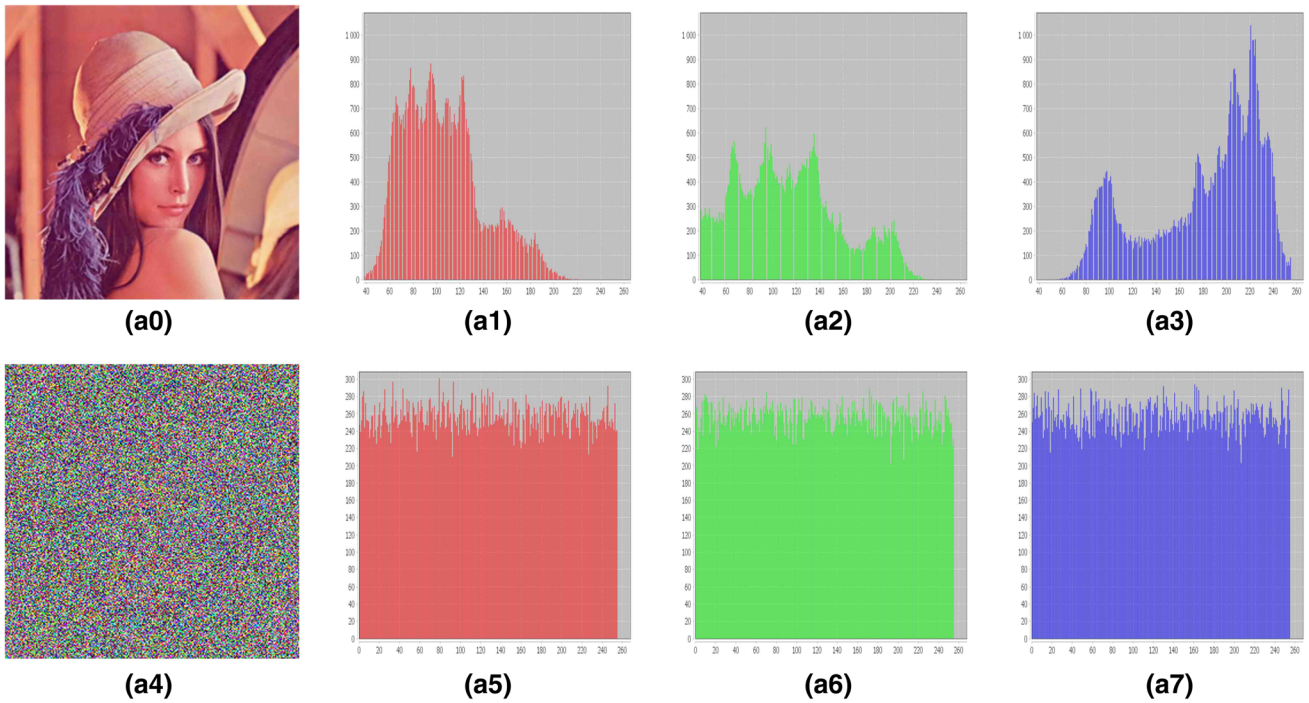


Fig. 8 **a0** Lena image, **a1** Red histogram of Lena image, **a2** Green histogram of Lena image, **a3** Blue histogram of Lena image, **a4** Encrypted Lena image, **a5** Red histogram of encrypted Lena image, **a6** Green histogram of encrypted Lena image, **a7** Blue histogram of encrypted Lena image (color figure online)

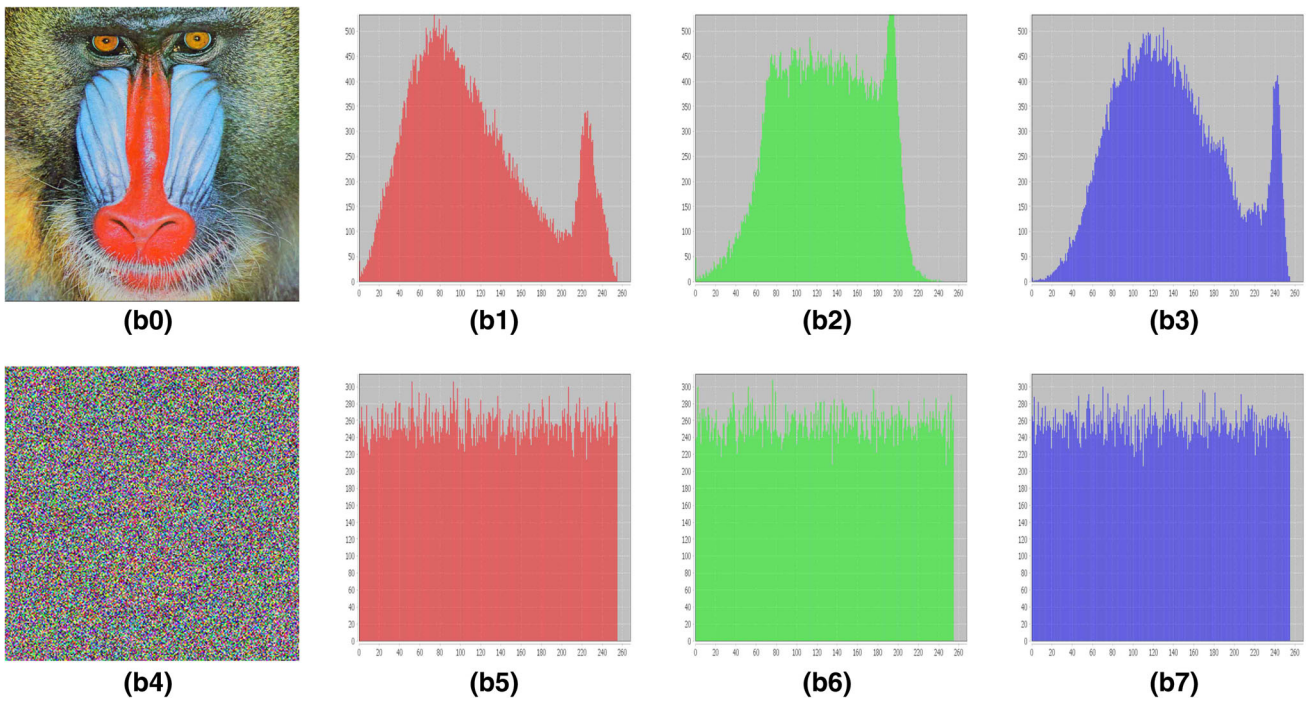


Fig. 9 **b0** Baboon image, **b1** Red histogram of Baboon image, **b2** Green histogram of Baboon image, **b3** Blue histogram of Baboon image, **b4** Encrypted Baboon image, **b5** Red histogram of encrypted Baboon image, **b6** Green histogram of encrypted Baboon image, **b7** Blue histogram of encrypted Baboon image (color figure online)

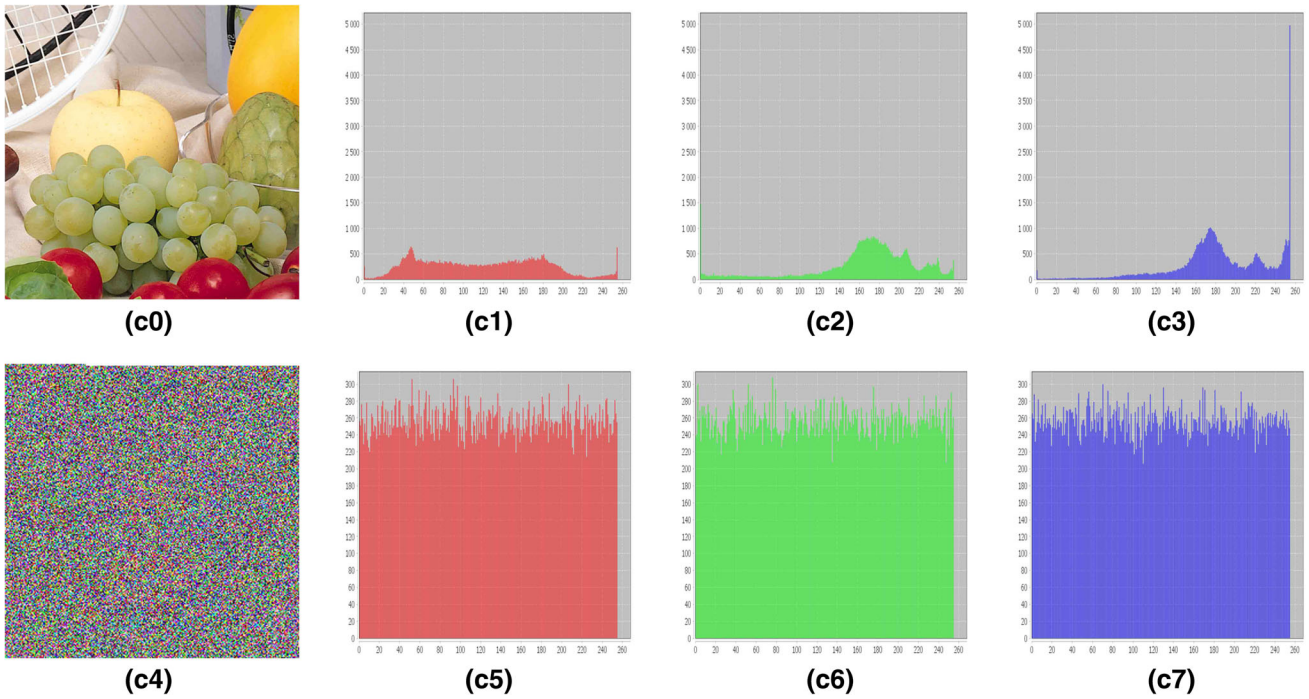


Fig. 10 **c0** Fruits image, **c1** Red histogram of Fruits image, **c2** Green histogram of Fruits image, **c3** Blue histogram of Fruits image, **c4** Encrypted Fruits image, **c5** Red histogram of encrypted Fruits image, **c6** Green histogram of encrypted Fruits image, **c7** Blue histogram of encrypted Fruits image (color figure online)

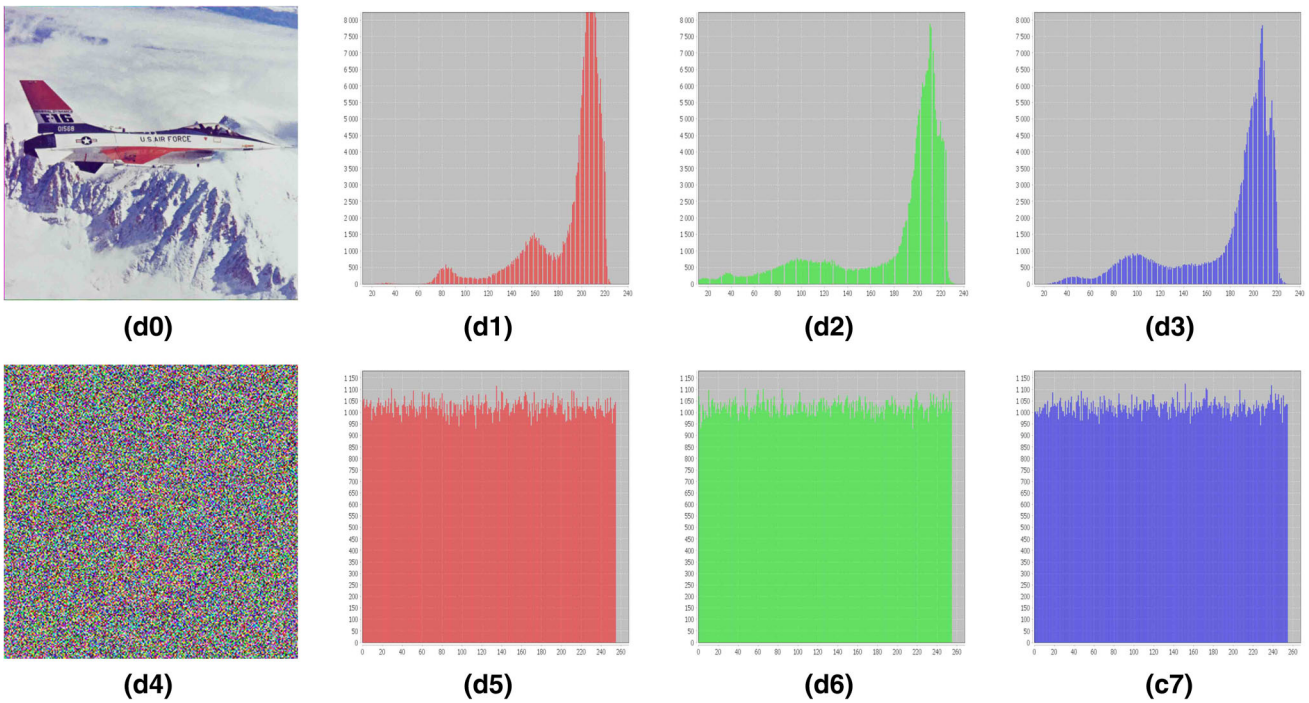


Fig. 11 **d0** Airplane image, **d1** Red histogram of Airplane image, **d2** Green histogram of Airplane image, **d3** Blue histogram of Airplane image, **d4** Encrypted Airplane image, **d5** Red histogram of encrypted Airplane image, **d6** Green histogram of encrypted Airplane image, **d7** Blue histogram of encrypted Airplane image (color figure online)

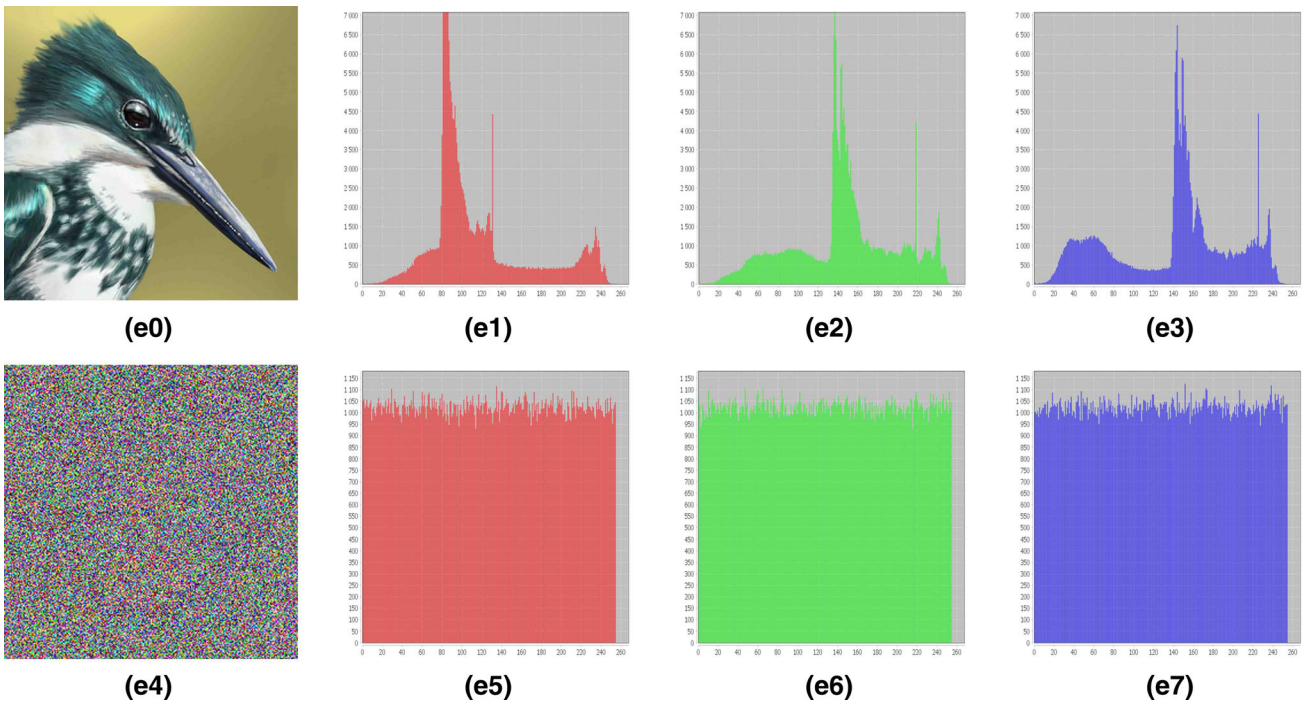


Fig. 12 **e0** Bird image, **e1** Red histogram of Bird image, **e2** Green histogram of Bird image, **e3** Blue histogram of Bird image, **e4** Encrypted Bird image, **e5** Red histogram of encrypted Bird image, **e6** Green histogram of encrypted Bird image, **e7** Blue histogram of encrypted Bird image (color figure online)

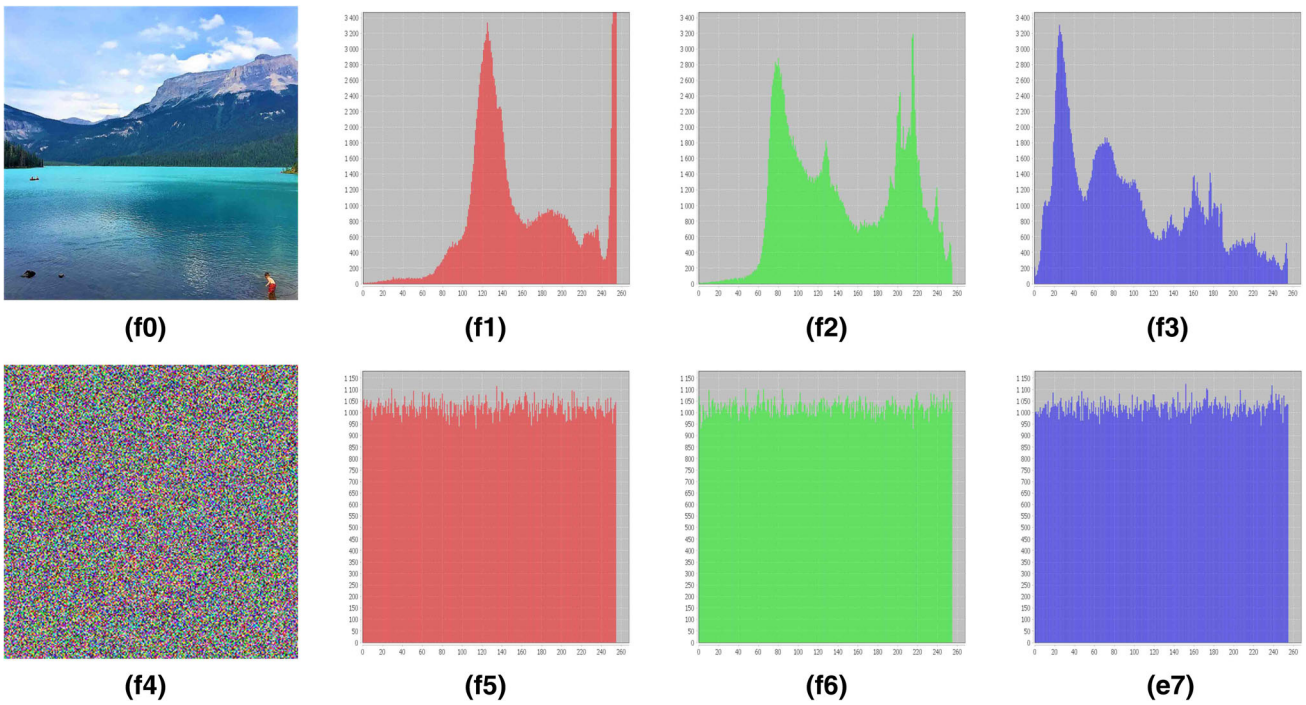


Fig. 13 **f0** Lake image, **f1** Red histogram of Lake image, **f2** Green histogram of Lake image, **f3** Blue histogram of Lake image, **f4** Encrypted Lake image, **f5** Red histogram of encrypted Lake image, **f6** Green histogram of encrypted Lake image, **f7** Blue histogram of encrypted Lake image (color figure online)

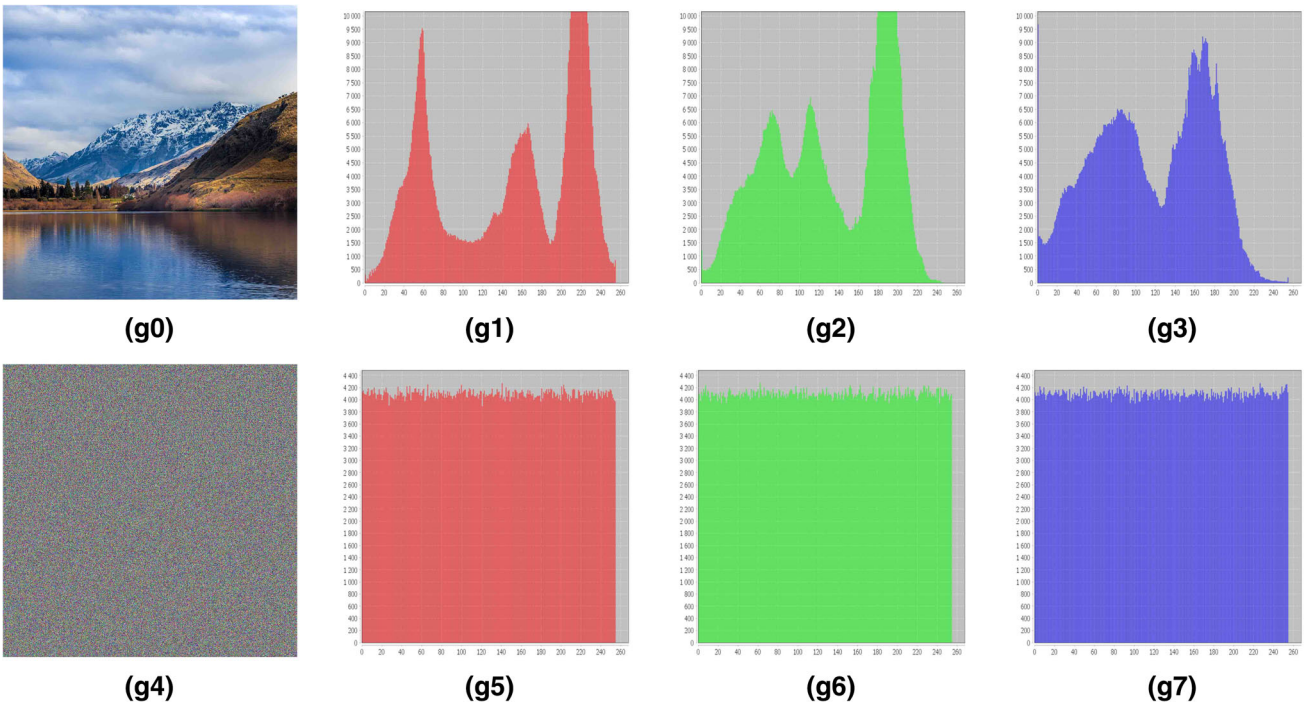


Fig. 14 **g0** Mountain image, **g1** Red histogram of Mountain image, **g2** Green histogram of Mountain image, **g3** Blue histogram of Mountain image, **g4** Encrypted Mountain image, **g5** Red histogram of encrypted Mountain image, **g6** Green histogram of encrypted Mountain image, **g7** Blue histogram of encrypted Mountain image (color figure online)

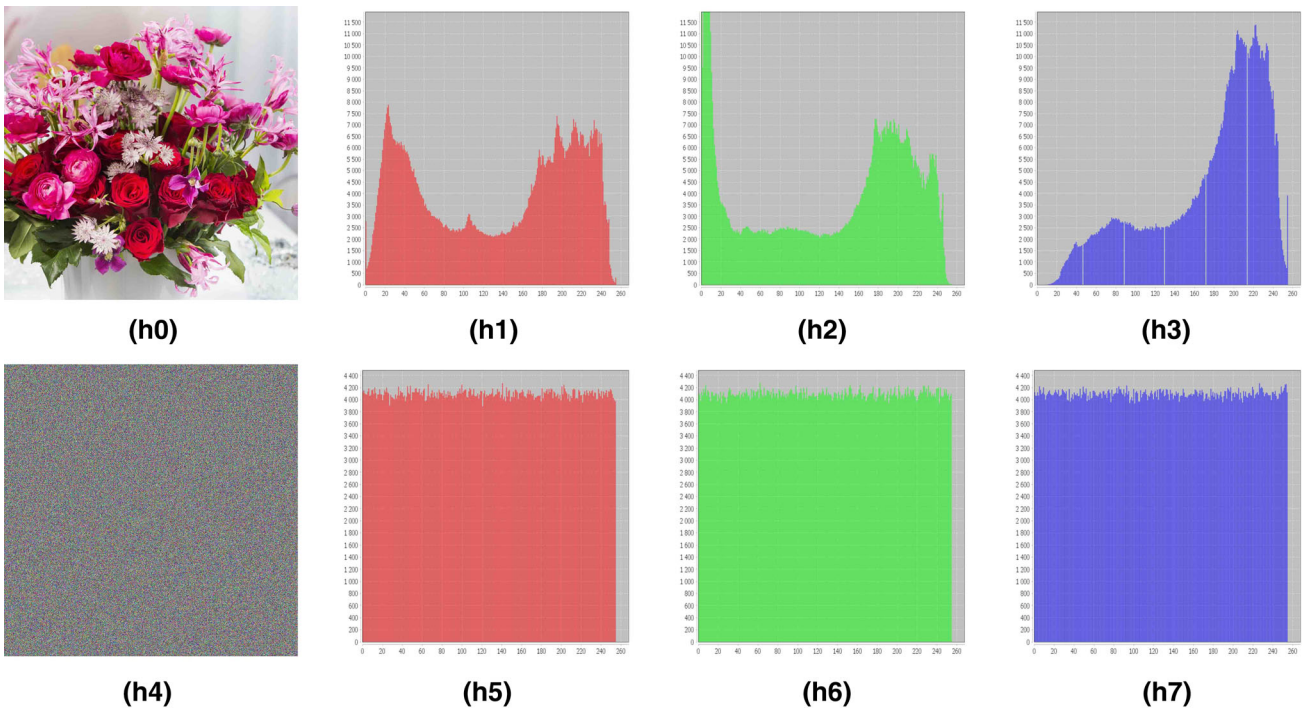


Fig. 15 **h0** Flowers image, **h1** Red histogram of Flowers image, **h2** Green histogram of Flowers image, **h3** Blue histogram of Flowers image, **h4** Encrypted Flowers image, **h5** Red histogram of encrypted Flowers image, **h6** Green histogram of encrypted Flowers image, **h7** Blue histogram of encrypted Flowers image (color figure online)

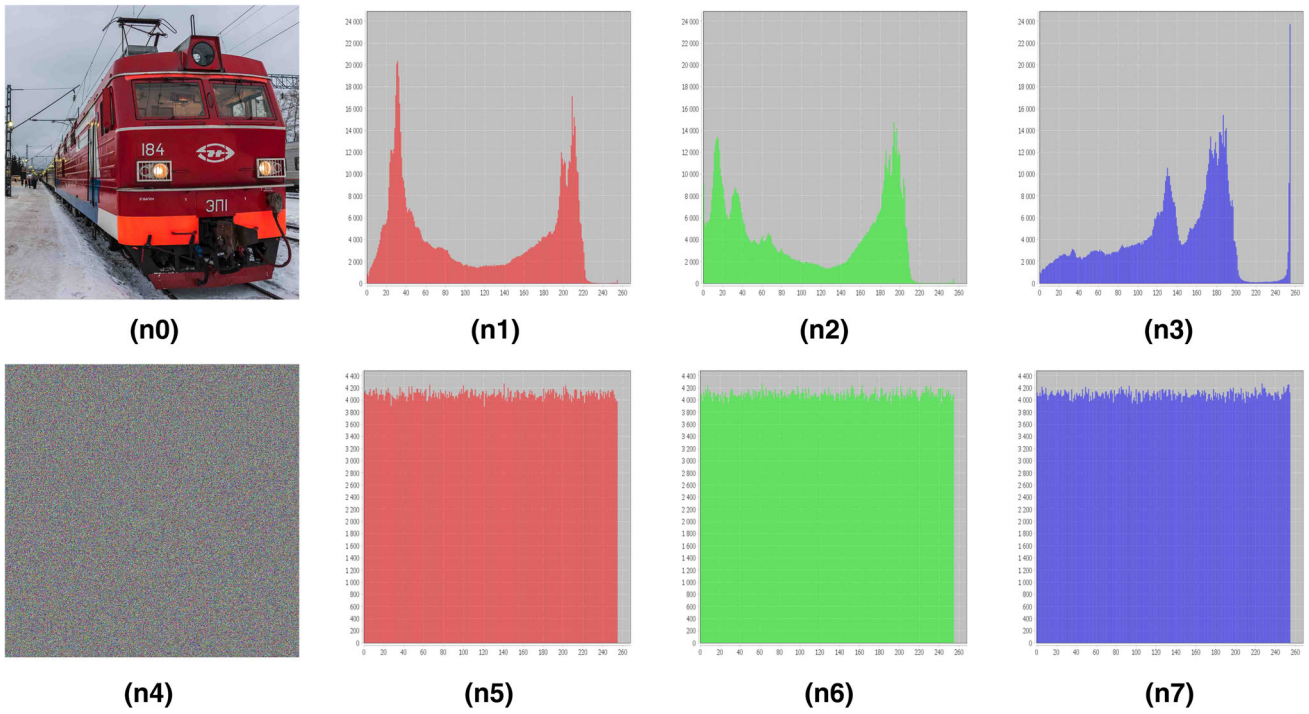


Fig. 16 **n0** Train image, **n1** Red histogram of Train image, **n2** Green histogram of Train image, **n3** Blue histogram of Train image, **n4** Encrypted Train image, **n5** Red histogram of encrypted Train image, **n6** Green histogram of encrypted Train image, **n7** Blue histogram of encrypted Train image (color figure online)

encrypted image must be widely different than the other one before the change. To analyze the resistance of our approach against this type of attack, we measured the number of pixels changed by calculating the NPCR (Number of Changing Pixel Rate) and the UACI (Unified Averaged Changed Intensity) by applying the following formulas:

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{11}$$

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{12}$$

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \tag{13}$$

With M and N, respectively, representing the height and width of the encrypted image, C_1 and C_2 are the encrypted images with a small difference at one pixel for each channel (red, green and blue) of the original image. To say that our approach is resistant to differential attacks, the NPCR and UACI measures must be large enough. Table 5 shows the values obtained by applying our approach on Lena image and Baboon image and compared them with those obtained by the Norouzi method (Norouzi et al. 2013) and Tong method (Tong et al. 2015).

To say that a method is resistant against differential attacks, the values of NPCR and UACI must be, respectively, greater than 99.6% and 33.4%; these values are justified (Es-Sabry et al. 2018b) by Eqs. (14) and (15).

$$\mathrm{NPCR}_E = (1 - 2^{-n}) \times 100\% \tag{14}$$

$$\mathrm{UACI}_E = \frac{1}{2^{2n}} \frac{\sum_{i=1}^{2^n - 1} i(i+1)}{2^n - 1} \times 100\% \tag{15}$$

Where n represent the number of bits needed to encode a pixel, which is coded on 8 bits, since the calculation is done for each channel (red, green and blue). And therefore, $\mathrm{NPCR}_E = 99.6094\%$ and $\mathrm{UACI}_E = 33.4635\%$.

The same thing for the key, if a small change made in the key (Matrix1, Matrix2), that will completely change the encrypted image; so we can say that the encryption system is sensitive to the change of the key. In our case, we used the Lena image with size of 256×256, and it was encrypted with the two keys (Matrix1, Matrix2) (Fig. 4). Then, we took the first matrix and we switched the last two values located at the right bottom of our matrix (2, 81); then, we encrypted the Lena image with the new keys (Matrix3 which is Matrix1 modified and Matrix2).

After calculating the NPCR and UACI measurements, we notice that the results obtained by our method are good (Table 6) and better than those obtained by Tong et al.

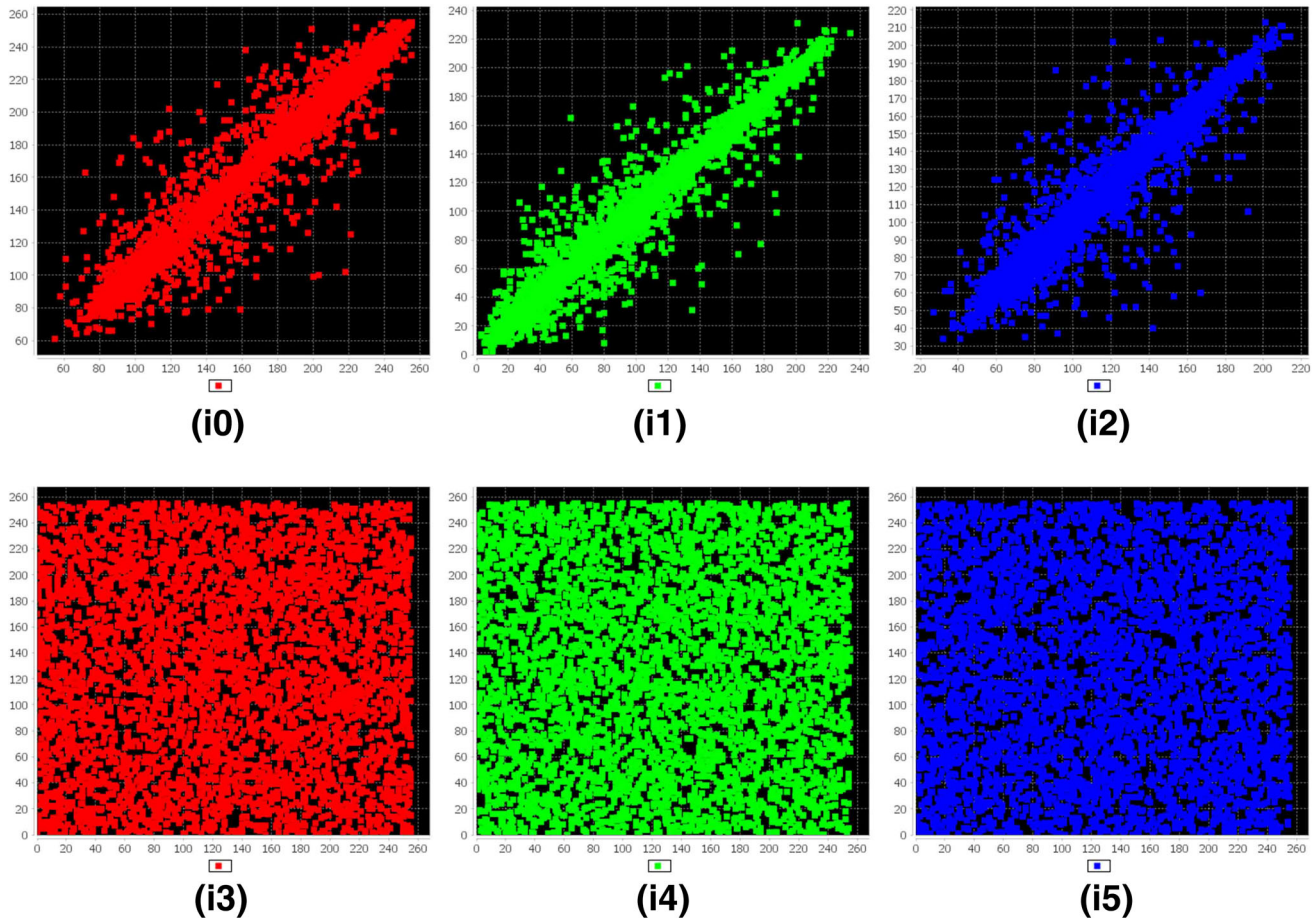


Fig. 17 Pixels distribution in the horizontal direction of Lena image: **i0** Original red channel, **i1** Original green channel, **i2** Original blue channel, **i3** Encrypted red channel, **i4** Encrypted green channel, **i5** Encrypted blue channel (color figure online)

(2015), which means our method is more efficient than Thong method.

Figure 20 shows the decryption result of the Lena image with the initial key (Matrix1, Matrix2) and with the modified key (Matrix3, Matrix2). After making a small change at a single pixel of Matrix1, we notice that the decryption operation failed completely.

2.5 Entropy analysis

Entropy is one of the most notable features for measuring the randomness of the image encryption algorithm. To calculate entropy information, we use the following formula.

$$H(s) = \sum_{i=0}^{2^n-1} P(s_i) \log_2[P(s_i)] \tag{16}$$

With $P(s_i)$ is the probability of occurrence for each s_i, 2^n is the total number of states of the information source. For a perfect random information source with 2^n states, the entropy should be equal to n.

In our case, the computation is done for each channel (red, green and blue), so the number of states is $2^n = 256$, and consequently an ideal entropy is $n = 8$.

Table 7 shows the different entropy values obtained for each of Figs. 8, 9, 10 and 11. These values are very close to the ideal value 8, which means that our method is resistant against the entropy attacks.

2.6 MSE and peak signal-to-noise ratio analysis

The last two parameters to evaluate the reliability of our algorithm are the MSE (Mean Square Error) and the PSNR (Peak Signal-to-Noise Ratio). The first criterion (MSE)

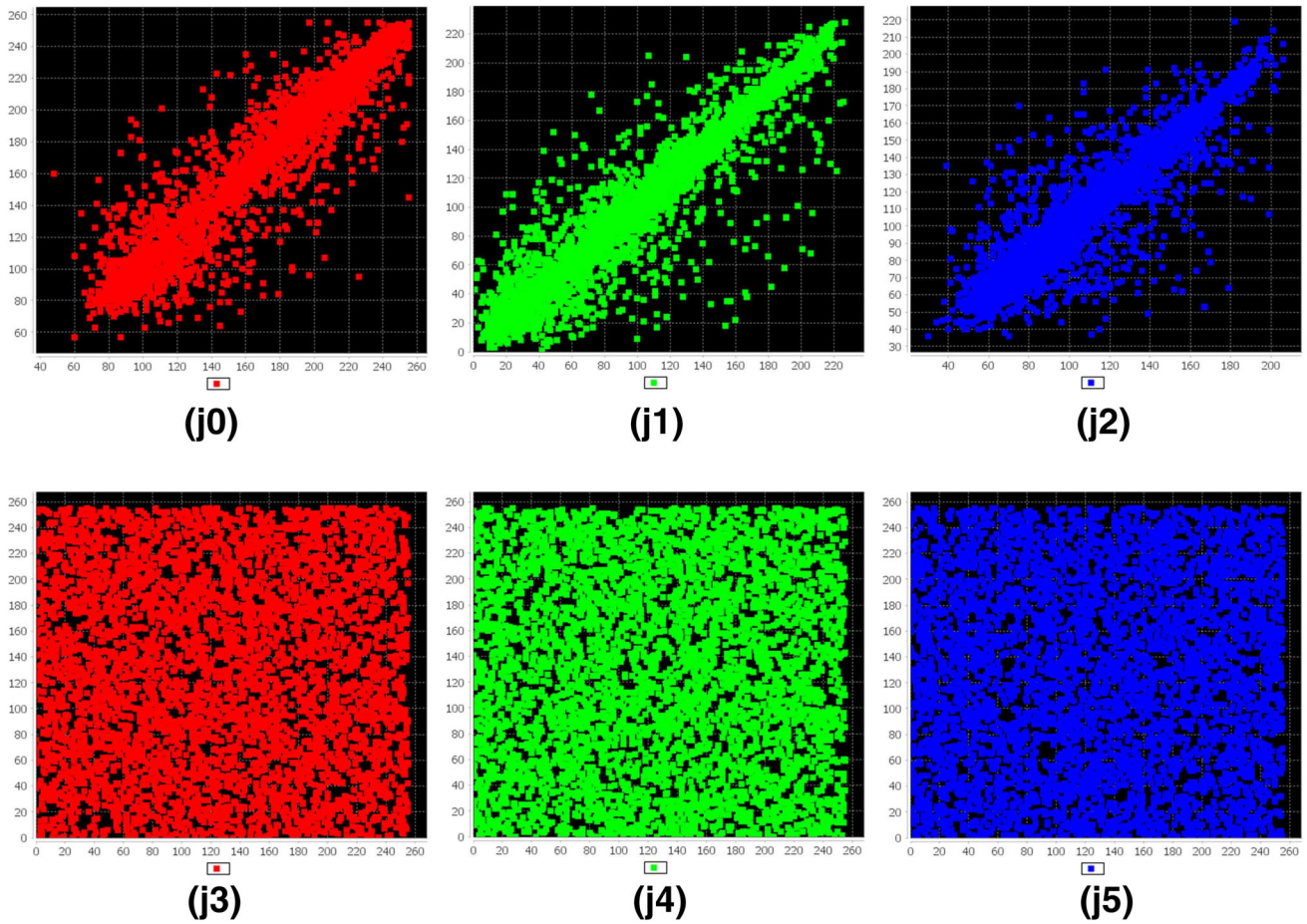


Fig. 18 Pixels distribution in the vertical direction of Lena image: **j0** Original red channel, **j1** Original green channel, **j2** Original blue channel, **j3** Encrypted red channel, **j4** Encrypted green channel, **j5** Encrypted blue channel (color figure online)

Table 2 Correlation coefficient of original images and encrypted images

Image	Channel	Original image			Encrypted image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	Red	0.975772	0.952779	0.918953	− 0.02093	0.048895	0.017922
	Green	0.970801	0.941692	0.935441	0.00813	0.010824	− 0.012458
	Blue	0.940875	0.913843	0.882293	− 0.008816	0.017843	− 0.034915
	Average	0.962483	0.936105	0.912229	− 0.007205	0.025854	− 0.009817
Baboon	Red	0.867365	0.918216	0.853941	− 0.024236	0.019584	0.017811
	Green	0.778304	0.805673	0.704966	0.002682	− 0.015062	− 0.010772
	Blue	0.860251	0.872245	0.805728	0.011979	− 0.008435	0.021916
	Average	0.835307	0.865378	0.788212	− 0.003192	− 0.001304	0.009651
Fruits	Red	0.934084	0.918770	0.877011	0.063041	0.038538	0.019305
	Green	0.935958	0.955326	0.909279	0.009570	0.007922	− 0.016128
	Blue	0.944211	0.953427	0.912598	0.017280	0.016386	0.005465
	Average	0.938084	0.942507	0.899629	0.029963	0.020948	0.002880

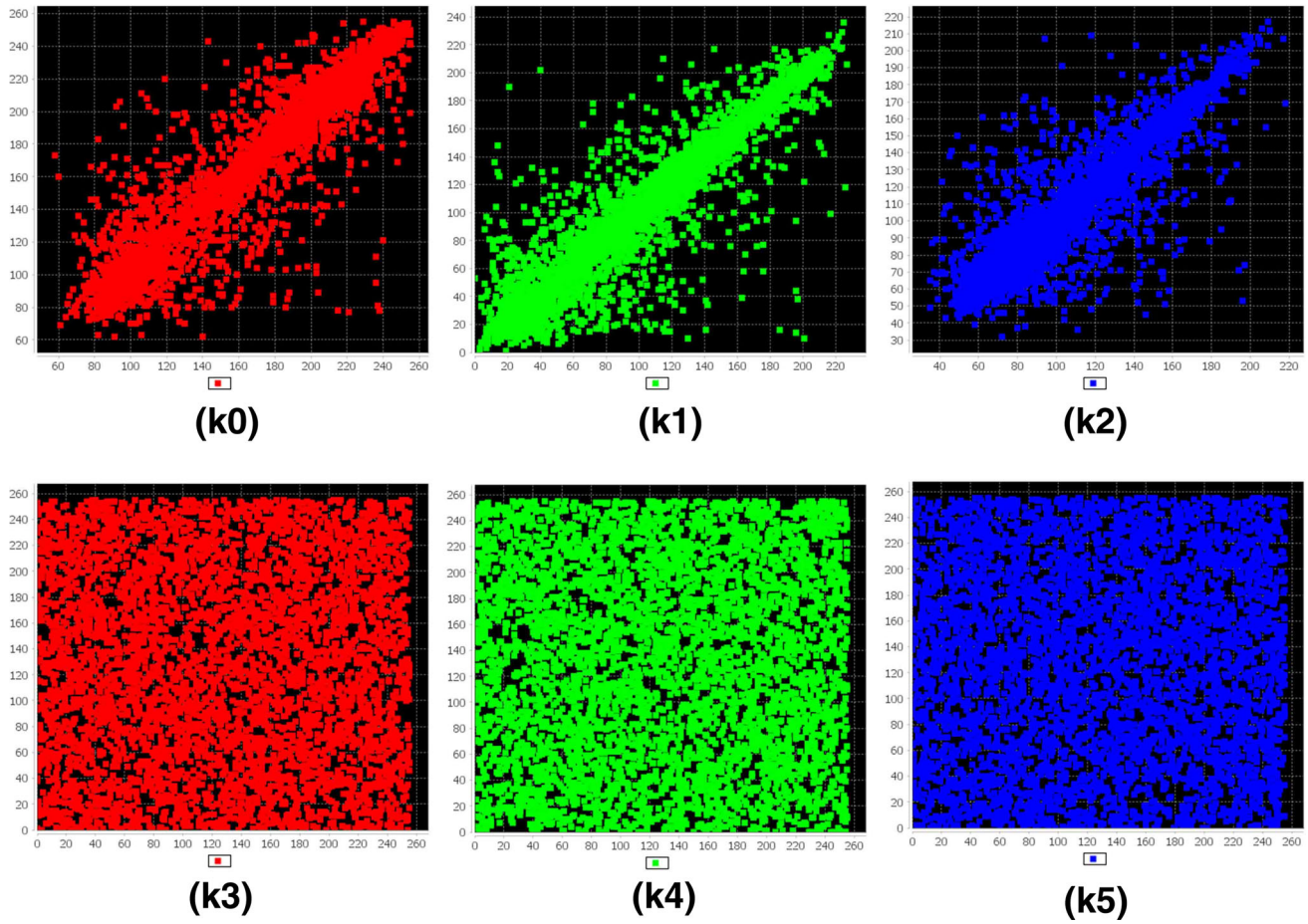


Fig. 19 Pixels distribution in the diagonal direction of Lena image: **k0** Original red channel, **k1** Original green channel, **k2** Original blue channel, **k3** Encrypted red channel, **k4** Encrypted green channel, **k5** Encrypted blue channel (color figure online)

gives the margin of error between the original image and the encrypted image and is calculated using the following formula.

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(A_{ij} - B_{ij}\right)^2 \tag{17}$$

With A represents the original image, B represents the encrypted image, M and N, respectively, represent the height and width of the images A and B. The security of the encrypted image increases with an increase in the MSE value.

The encrypted image quality is evaluated using the second criterion (PSNR), which is calculated using the following formula.

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{I_{\mathrm{Max}}^2}{\mathrm{MSE}}\right) \tag{18}$$

Where I_{Max} represents the maximum value of a pixel of the image. The PSNR value should be small, which means a big difference between the original image and the encrypted image.

The calculations are made for the four images in Figs. 8, 9, 10 and 11. Table 7 represents the values obtained for each channel (red, green and blue). From the results presented in this table, we notice that the values of the MSE are very large. On the other hand, the PSNR values are very small. So, we can say that our approach gives good results, which shows that the encrypted image is well secured.

After performing the decryption operation, the calculations were made again to evaluate the decryption quality by also calculating the CC (Correlation Coefficient) for each image and for each channel (red, green and blue). The MSE is 0, the CC is 1 and the PSNR tends to infinity (Table 7), which means that the decrypted image and the original

Table 3 CC comparison of two adjacent pixels of our method with other methods

Encrypted image	Directions			Average
	Horizontal	Vertical	Diagonal	
Tong et al. (2009)	0.017188	0.009852	0.033045	0.020028
Borujeni and Eshghi (2011)	0.004100	0.030800	0.005300	0.013400
Wang and Guo (2014)	0.006300	0.006200	0.006900	0.006466
Hua et al. (2015)	0.002383	0.008576	0.040242	0.017067
Tong et al. (2015)	0.003800	0.005800	0.013300	0.007633
Proposed method	− 0.007205	0.025854	− 0.009817	0.002944

Table 4 CC comparison of our method with ZHU method

Encrypted image	Channel			Average	Zhu (2012)
	Red	Green	Blue		
Lena	0.007621	0.005257	− 0.007645	0.001744	0.002851
Baboon	0.002012	0.007464	0.008313	0.005929	− 0.006632
Fruits	0.001015	− 0.006242	− 0.010838	− 0.005355	–
Peppers	0.014940	− 0.020151	0.002337	− 0.001332	− 0.001650
Bird	0.004091	0.000611	0.012817	0.005839	–
Airplane	0.004597	0.007441	0.006604	0.006214	–
Flowers	− 0.010705	0.012209	0.001583	0.001029	–
Lake	− 0.009461	0.014982	− 0.007487	− 0.000655	–

Table 5 Sensitivity of our proposed method for a single pixel change of the original image

Image	Channel	Proposed method		Norouzi et al. (2013)		Tong et al. (2015)	
		NPCR	UACI	NPCR	UACI	NPCR	UACI
Lena	Red	99.662780	33.440426	–	–	99.571230	33.341510
	Green	99.686486	33.523906	–	–	99.618720	33.352270
	Blue	99.678039	33.830632	–	–	99.692230	33.358140
	Average	99.675768	33.598321	99.668900	33.556100	99.627393	33.350640
Baboon	Red	99.653625	33.555644	–	–	–	–
	Green	99.630737	33.736075	–	–	–	–
	Blue	99.636159	33.577522	–	–	–	–
	Average	99.640173	33.623080	99.638400	33.630500	–	–

Table 6 Sensitivity of our proposed method for a single pixel change of one of the keys

Image	Channel	Proposed method		Tong et al. (2015)	
		NPCR	UACI	NPCR	UACI
Lena	Red	99.719238	33.694923	99.404760	33.224270
	Green	99.589538	33.231913	99.330560	33.334510
	Blue	99.777221	33.702512	99.484530	33.289340
	Average	99.695332	33.543116	99.406616	33.282706

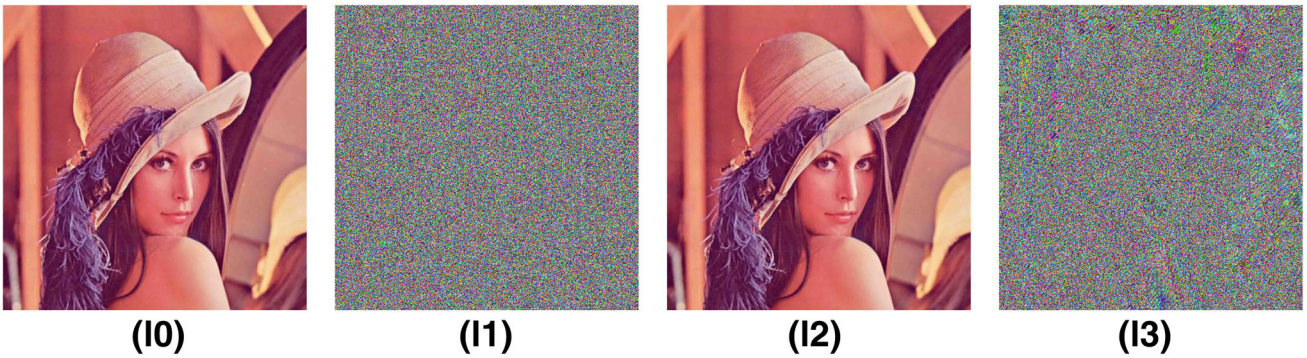


Fig. 20 **I0** Lena image, **I1** Encryption of Lena Image with key (Matrix1, Matrix2), **I2** Decryption of Lena Image with key (Matrix1, Matrix2), **I3** Decryption of Lena Image with key (Matrix3, Matrix2) (color figure online)

Table 7 Parameters of the encryption and decryption quality

Image	Channel	Encryption			Decryption		
		MSE	PSNR	Entropy	MSE	PSNR	CC
Lena	Red	10442.20	7.942880	7.997080	0	inf	1
	Green	8756.56	8.034901	7.997886	0	inf	1
	Blue	6878.10	8.513158	7.997364	0	inf	1
	Average	8692.29	8.163647	7.997443	0	inf	1
Baboon	Red	8283.85	8.948480	7.997216	0	inf	1
	Green	7556.56	9.347557	7.997747	0	inf	1
	Blue	9134.69	8.523861	7.997340	0	inf	1
	Average	8325.03	8.939966	7.997434	0	inf	1
Fruits	Red	10701.97	7.836163	7.999871	0	inf	1
	Green	9559.38	8.326501	7.996846	0	inf	1
	Blue	9029.87	8.573987	7.992466	0	inf	1
	Average	9763.74	8.245550	7.996394	0	inf	1
Peppers	Red	7809.55	8.194256	7.999087	0	inf	1
	Green	10642.07	7.583677	7.990610	0	inf	1
	Blue	10490.70	6.988663	7.991868	0	inf	1
	Average	9647.44	7.588865	7.993855	0	inf	1

image are identical. Therefore, we can conclude that our method gives a high precision.

3 Conclusion

In this paper, we proposed a new method that encrypts color images in a very secure way. The method is based on the random numbers generation of two matrices with size of 16×16 containing different integers between 0 and 255. Then, we applied a right circular shift according to a function that takes into account the values of the two matrices as well as their position (row and column), the original image pixels and the keys values. The resistance of our method against statistical attacks, differential attacks and entropy attack has been proved. Our method has great sensitivity in terms of a single pixel change of the original image as well as the values of one of the two keys; the NPCR and UACI measurements justify this sensitivity. The decryption operation is similar in a reverse manner to that of encryption; the decryption quality is perfect and is justified by the measures of CC, MSE and PSNR.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Arroyo D, Rhouma R, Alvarez G, Li S, Fernandez V (2008) On the security of a new image encryption scheme based on chaotic

map lattices. Chaos Interdiscip J Nonlinear Sci. https://doi.org/10.1063/1.2959102
Borujeni SE, Eshghi M (2011) Chaotic image encryption system using phase-magnitude transformation and pixel substitution. J Telecommun Syst 52:525–537
Es-Sabry M, El Akkad N, Merras M, Saaidi A, Satori K (2018a) A novel text encryption algorithm based on the two-square Cipher and Caesar Cipher. Int Conf Big Data Cloud Appl 872:78–88
Es-Sabry M, El Akkad N, Merras M, Saaidi A, Satori K (2018b) Grayscale image encryption using shift bits operations. Int Conf Intell Syst Comput Vis. https://doi.org/10.1109/ISACV.2018.8354028
Fridrich J (1997) Image encryption based on chaotic maps. In: 1997 IEEE international conference on systems, man, and cybernetics. Computational cybernetics and simulation, vol 2. pp 1105–1110
Hua ZY, Zhou YC, Pun CM, Philip Chen CL (2015) 2D Sine logistic modulation map for image encryption. Inf Sci 297:80–94
Hwang DC, Shin DH, Kim ES (2007) A novel three-dimensional digital watermarking scheme basing on integral imaging. Optics Commun 277:40–49
Lesne A (2006) Chaos in biology. Riv Biol 99:467–481
Li C, Li S, Asim M, Nunez J, Alvarez G, Chen G (2009) On the security defects of an image encryption scheme. Image Vis Comput 27:1371–1381
Ling C, Wu X, Sun S (1999) A general efficient method for chaotic signal estimation. IEEE Trans Signal Process 47:1424–1428
Liu CX, Lu JJ (2010) A novel fractional order hyperchaotic system and its circuit realization. Int J Mod Phys B 24:1299–1307
Lu L, Luan L, Meng L, Li CR (2012) Study on spatiotemporal chaos tracking synchronization of a class of complex network. Nonlinear Dyn 70:89–95
Norouzi B, Seyedzadeh SM, Mirzakuchaki S, Mosavi MR (2013) A novel image encryption based on hash function with only two-round diffusion process. Multimed Syst 20:45–64
Piao YR, Shin DH, Kim ES (2009) Robust image encryption by combined use of integral imaging and pixel scrambling techniques. Opt Lasers Eng 47:1273–1281
Singh N, Sinha A (2008) Optical image encryption using fractional Fourier transform and chaos. Opt Lasers Eng 46:117–123
Skrobek A (2007) Cryptanalysis of chaotic stream cipher. Phys Lett A 363:84–90
Tong X, Cui M, Wang Z (2009) A new feedback image encryption scheme based on perturbation with dynamical compound chaotic sequence cipher generator. J Opt Commun 282:2722–2728
Tong XG, Wang Z, Zhang M, Liu Y, Xu H, Ma J (2015) An image encryption algorithm based on the perturbed high-dimensional chaotic map. Nonlinear Dyn 80:1493–1508
Wang XY, Guo K (2014) A new image alternate encryption algorithm based on chaotic map. Nonlinear Dyn 76:1943–1950
Wang X, Zhao D, Chen L (2006) Image encryption based on extended fractional Fourier transform and digital holography technique. Opt Commun 260:449–453
Weidenmuller HA, Mitchell GE (2009) Random matrices and chaos in nuclear physics. Rev Mod Phys 81:539–589
Wu X, Hu H, Zhang B (2004) Parameter estimation only from the symbolic sequences generated by chaos system. Chaos Solitons Fractals 22:359–366
Zhang L, Liao X, Wang X (2005) An image encryption approach based on chaotic maps. Chaos Solitons Fractals 24:759–765
Zhu C (2012) A novel image encryption scheme based on improved hyperchaotic sequences. J Opt Commun 285:29–37