Hindawi
Complexity
Volume 2022, Article ID 3865820, 22 pages
https://doi.org/10.1155/2022/3865820

Research Article

A Simple Image Encryption Based on Binary Image Affine Transformation and Zigzag Process

Adélaïde Nicole Kengnou Telem,[1,2] **Cyrille Feudjio**,[1] **Balamurali Ramakrishnan**,[3] **Hilaire Bertrand Fotsin**,[2] **and Karthikeyan Rajagopal**[3]

[1]*Department of Electrical and Electronic Engineering, College of Technology (COT), University of Buea, P.O. Box 63, Buea, Cameroon*
[2]*Laboratoire de recherche de Matière Condensée, d'Electronique et de Traitement du Signal (LAMACETS), Département de Physique, Faculté des Sciences, Université de Dschang, Dschang, Cameroon*
[3]*Centre for Nonlinear Systems, Chennai Institute of Technology, Chennai, India*

Correspondence should be addressed to Adélaïde Nicole Kengnou Telem; adelkengnou@gmail.com

Received 20 April 2021; Revised 23 September 2021; Accepted 19 November 2021; Published 7 January 2022

Academic Editor: Padmapriya Praveenkumar

In this paper, we propose a new and simple method for image encryption. It uses an external secret key of 128 bits long and an internal secret key. The novelties of the proposed encryption process are the methods used to extract an internal key to apply the zigzag process, affine transformation, and substitution-diffusion process. Initially, an original gray-scale image is converted into binary images. An internal secret key is extracted from binary images. The two keys are combined to compute the substitution-diffusion keys. The zigzag process is firstly applied on each binary image. Using an external key, every zigzag binary image is reflected or rotated and a new gray-scale image is reconstructed. The new image is divided into many nonoverlapping subblocks, and each subblock uses its own key to take out a substitution-diffusion process. We tested our algorithms on many biomedical and nonmedical images. It is seen from evaluation metrics that the proposed image encryption scheme provides good statistical and diffusion properties and can resist many kinds of attacks. It is an efficient and secure scheme for real-time encryption and transmission of biomedical images in telemedicine.

1. Introduction

The amazing developments in the field of network communications during the past years have created a great requirement for secured image transmission over the Internet [1]. Image data, such as medical images, military images, images of electronic publishing, and fingerprint images from authentication systems, must be kept private and confidential. The confidentiality of these images is capital and cannot be guaranteed through the Internet. To ensure the security of information during transmission, encryption techniques are used. Cryptography aims to ensure data confidentiality, integrity, and authentication during communication. In telemedicine, medical data need to be processed with total discretion. This justifies the use of encryption technology in telemedicine.

Cryptographic methods are based on two fundamental principles, namely, substitution and diffusion. Substitution involves replacing certain letters or values of the original message with others. Diffusion consists of dispersing the position of the letters or the values of the original message to scramble the message. Most encryption techniques have been developed to secure text data. Unfortunately, these techniques are not suitable for images because the images have a rather complex structure and are quite large in size compared to the text data. Yet, the images may contain private and fairly confidential information. Hence, there is a need to find an effective technique to secure images. Thus, designing less complex image encryption algorithms becomes very crucial. Much research has focused on the issue of image security. Digital images have several properties such as information redundancy, a strong correlation between pixels, and large data capacities. An image encryption algorithm must take into account all these properties.

Conventional image encryption techniques are divided into two groups, namely, asymmetric encryption (with private and public keys) and symmetric encryption (with a secret key) [2]. Several algorithms have been proposed in the past for the encryption of images. We have data encryption standard (DES), triple data encryption algorithm (TDEA), advanced encryption standard (AES), Rivest, Shamir, and Adleman (RSA), and fast image encryption algorithm (FEAL) [2–8]. In recent years, chaotic systems have been used by many researchers in image encryption. A chaos-based image encryption technique typically uses both substitution and diffusion processes. This technique generally uses an external key and one or more generators to generate chaotic sequences that will be used for the substitution/diffusion process. Ahmad and Farooq in [9] approved that the generation of high-quality key stream decides the level of security offered by the Cipher. They combined the simple logistic map with the cubic map to generate PN sequences for the proposed encryption scheme. Those PN sequences from the proposed generator have good autocorrelation and have been tested randomness. Li et al. in [10] introduced a performance-enhanced image encryption scheme based on depth-conversion integral imaging and hybrid cellular automata (CA). The aim is to meet the requirements of secured image transmission. The input image is firstly decomposed into an elemental image array (EIA) using the depth-converted integral imaging technique. The CA model and chaotic sequence are used to encrypt the elemental images. In [11], Ahmad et al. used particle swarm optimization and chaotic map to propose an optimized image encryption. Initial conditions of the chaotic map depend on the pending plain image, so the algorithm is resistant because, from one image to another, key stream will be different. Niu et al. in [12] proposed an efficient method for image encryption based on the chaos theory and a deoxyribonucleic acid (DNA) sequence database using the characteristics of chaos, such as randomness, regularity, ergodicity, and initial value sensitiveness, combined with the unique space conformation of DNA molecules and their unique information storage and processing ability. In [13], Kengnou et al. proposed an encryption algorithm using a 3D chaotic system and DNA coding. Two keys are used, an internal one and an external one. During the chaotic process, the sequences generated from the different variables of the chaotic generator are not used separately. They are combined using the zigzag process and used simultaneously. Each 3D chaotic system can be used. Twenty four DNA rules and 16 join operations of DNA coding are used during the DNA coding process. A Fast Walsh Transform (FWT) has been combined with two chaotic logistic maps by Telem et al. in [14] to propose a new scheme of image encryption. Chaotic encryption methods are combined with the two-dimensional FWT of images. Liu and Miao in [15] proposed an image encryption method based on a logistic chaotic map and dynamical algorithm. In their method, the parameter of the logistic map is varied and used to shuffle the plain image. Then, the dynamical algorithm comes to encrypt the image. Many other chaotic encryption methods have been proposed in [15–23].

Cryptanalysis evaluates the efficiency of cryptosystems to resist against attacks and therefore confirms their validity. Several studies have shown that some cryptosystems based on chaos could be cryptanalyzed [24–27]. To face this challenge, Pareek et al. in [28] proposed an image encryption system without chaos. From a 128-bit external key, it provides an efficient algorithm using 16 rounds with satisfactory results. In [29], Jolfaei et al. have demonstrated the shortcomings of the image encryption scheme proposed by Pareek et al. [28]. The method is not secured, and the secret key can be deduced by a chosen-ciphertext attack. Security flaws of the encryption scheme have been discussed and solutions have been proposed in [29]. Houas et al. proposed in [30] a new algorithm to encrypt binary images based on several steps. Firstly, they reduced the amount of data required to present the image. The next step consists to divide the image into d blocks. Those blocks are used to construct a new image of the same size as the original one but represented on a new basis. The construction of the new basis is inspired by the work of Mokhtari and Melkemi [31]. After that transformation, they obtained a key image and used it to encrypt and decrypt images. The encrypted image is the representation on a new basis. The decryption algorithm consists of subtraction between each encrypted image and the key image and the sum of them.

Some image encryption methods are based on mathematical transform such as cosine transform, and Fourier transform is proposed. The method proposed by Lima et al. in [32] is based on the cosine number transform (CNT), a mathematical tool whose application requires modular arithmetic only. A CNT is very sensitive to changes in the vector being transformed, so 2 slightly different vectors may have significantly distinct CNTs, which are desirable for cryptography applications [32]. The method consists of dividing an image into blocks that overlap horizontally and vertically the corresponding adjacent block. The blocks are then sequentially taken and submitted to the recursive computation of a two-dimensional CNT. The method is limited to noncompressed images and particularly to medical images complying with the Digital Imaging and Communications in Medicine (DICOM) standard. Lima et al. in [33] proposed a fast computation of Cosine transform over fields of characteristic 2 and the application to image encryption. Annaby et al. in [34] have proposed a cryptosystem based on Fourier transform.

In [35–38], methods to reduce the workload of the time-consuming diffusion part are proposed. The encryption process is taken over the entire image. The image is not divided into many subblocks. Those methods gain in execution time but have several security problems.

In [39–41], affine transformations combined with other mathematical functions have been used to propose image encryption methods. Zhu et al. [39] have used affine cipher and generalized Arnold map to propose an image encryption scheme. Ahmad and Hwang in [40] have combined affine transformation with the chaotic map to propose their encryption method. In [41], Shah et al. have combined affine transformation with linear fractional transformation.

In this work, affine transformations are not combined with other mathematical models or functions. We propose a new image encryption algorithm using an external key, an internal key, affine transformations (reflections and rotations), zigzag process, and substitution-diffusion processes. Each pixel of the plain image is converted into its equivalent 8-bit binary, and the bits of rank n ($n = 1, 2, 3, 4, 5, 6, 7, 8$) of the different pixels constitute the binary image of this rank. So, the plain image is converted into 8 binary images. From binary images, an internal key is deduced. Each binary image is submitted to the zigzag process to yield zigzag binary images. Using an external key, the zigzag binary images are mapped using affine transformations and a new image is reconstructed. The reconstructed image is then submitted to substitution and diffusion processes to produce a cipher image. The novelties of this work are

(i) The use of the two keys: an internal key is extracted from the binary components of the plain image to be encrypted. In [13], an internal key is extracted directly from the pixel of plain image and not from the binary components of the plain image. In this work, an internal key is used in combination with the external key to generate the substitution and diffusion sequences. Internal keys are different from one plain image to another, and consequently, the substitution and diffusion sequences are also different even in the case of the same external key. To decrypt the cipher image, one must have the exact internal key and external key and well-known algorithm used in the cryptosystem.

(ii) The proposed algorithm does not use chaotic generators or complex mathematic functions to compute the substitution-diffusion sequences. Rather, those sequences are computed from simple logical bit operation by using external and internal keys and the part of the previous result of the encryption process.

(iii) The application of affine transformation on the binary component of the plain image.

(iv) The method to apply zigzag process: depending on the context, it can act as a diffusion or substitution process. Previous works have applied the zigzag process on the gray-scale image. The proposed method applied the zigzag process on the binary version of the plain image. So, it acts as a substitution process.

Hence, the proposed method does not gain on execution time but gain more on efficiency, security, and robustness. The proposed method is very helpful in telemedicine to secure medical images before transmitting them from one hospital to another. The method can also be used in every domain where we need to secure images as military domain and so on.

In the rest of the work, we present the details of our encryption algorithm in Section 2. The experimental results and security analysis tests are given in Section 3. At the end of the work is a conclusion.

2. Proposed Cryptosystem

2.1. Block Diagram of the Proposed Algorithm. In this work, we used an external key of 128 bits long, an internal key, reflection and rotation mappings, zigzag process, substitution, and diffusion processes to propose a new image encryption method. The plain image is converted into 8 binary images. From binary images, an internal key is extracted. Each binary image is submitted to the zigzag process to yield zigzag binary images. Using an external key, the zigzag binary images are mapped and a new image is reconstructed. This new image is divided into nonoverlapping squared subblocks. Each subblock is then submitted to the substitution and diffusion processes for K round using the combination of the two keys. The originality of this method dwells on the method used to generate the internal key, the use of reflection or rotation mapping, and the method to apply the substitution and the zigzag processes, the method used to compute substitution and diffusion sequences. The internal key is provided by the image being encrypted. In [28], the zigzag operation is applied on the pixel values, and it acts to change just the position of the pixel or the pixel location in the subblock. In this work, the whole image or the considered subblock is converted into binary images or binary subblock. The bits in a binary image/subblock are then reshuffled within the image/block by a zigzag path, and a new image/subblock is reconstructed. Consequently, the zigzag process changes the value of the pixels and acts as a substitution process. After the zigzag process, an external key is used to choose the corresponding type of transformation to be applied on each zigzag binary image. In [39–41], authors combine affine transformations with other mathematical models or functions. In this work, we do not use any chaotic generator or mathematical function to generate the codes used for the substitution-diffusion processes as usually. Rather, we combine the two keys (internal and external) and pixels of the image to compute the codes used on substitution-diffusion processes. Those features are the particularities of this algorithm. The block diagram of the proposed algorithm is shown in Figure 1.

2.1.1. External and Internal Keys. The proposed algorithm uses an external secret key of thirty-two hexadecimal numbers as shown in this example: «**ABCDEFGHIJKLMNOPRSTUVW*αβγηθλξρτφ***». This key is used for both substitution-diffusion processes and the choice of the type of affine transformation to be applied on binary images. An original image I_{mxn} is decomposed into 8 binary images Ibi_{mxn} ($i = 1, 2, \ldots, 8$). Ibi_{mxn} are submitted to "XOR" operation and yield one binary matrix. This matrix is then converted into a pixel matrix. The "XOR" operations between lines of the matrix produce the first part of the internal key named "keyi1," and while the "XOR" operations between columns produce the second part of the internal key named "keyi2." Table 1 presents the internal keys of two images 'ANTAMOEBACOLI' and 'Lena,' respectively. One can see the difference between the two internal keys. So, internal keys are different from image to another. "keyi1"

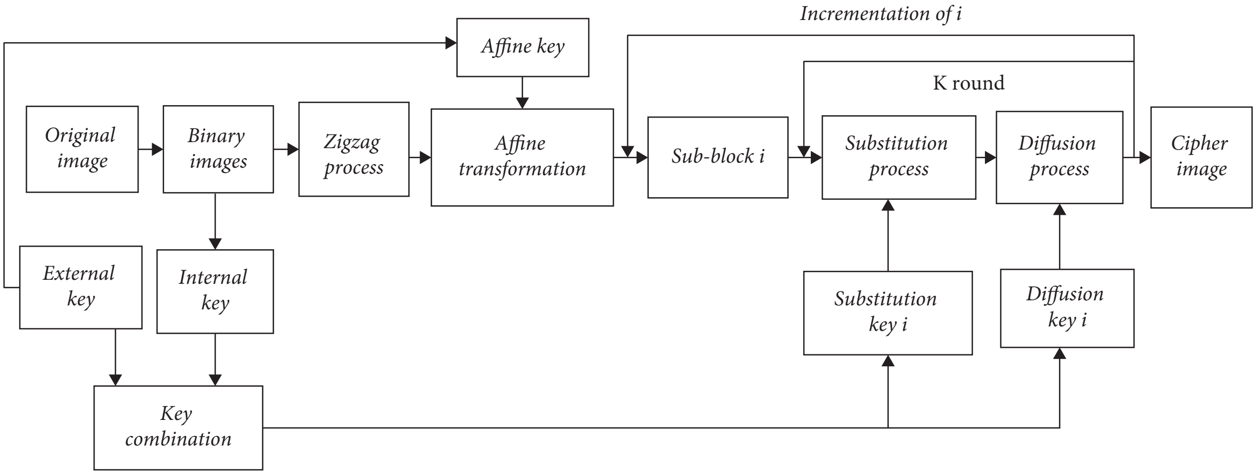


Figure 1: The block diagram of our proposed algorithm.

Table 1: Internal keys of two images: ANTAMOEBACOLI and 'Lena.'

78	188	117	104	223	128	20	173	163	149	88	177	175	200	20	226
129	69	76	140	81	116	102	113	169	172	48	18	59	53	221	31
65	64	217	252	193	182	106	40	4	45	101	69	209	255	63	181
92	174	248	130	243	120	29	157	150	246	46	245	134	62	174	142
158	34	176	210	11	152	156	2	78	50	54	71	55	220	66	248
45	194	164	89	70	238	221	204	56	57	58	43	223	104	67	249
143	234	255	50	51	52	53	54	60	61	207	64	65	68	70	72
55	56	57	58	59	60	61	62	73	74	75	76	77	79	80	81
Keyi1 of ANTAMOEBACOLI image								Keyi2 of ANTAMOEBACOLI image							
138	97	232	153	162	85	200	238	115	47	160	107	94	224	6	152
94	245	231	183	50	17	213	186	120	102	48	76	237	84	246	154
253	230	53	118	255	202	59	218	77	255	168	114	20	176	204	53
117	195	102	64	39	112	25	133	143	11	70	232	75	79	156	78
119	154	69	206	73	95	24	244	18	95	140	147	141	101	184	127
91	45	21	122	101	176	63	211	100	106	221	87	98	13	124	178
108	103	208	216	31	30	90	190	16	80	5	138	144	89	81	58
224	242	37	239	126	112	10	180	208	171	142	34	199	22	128	82
Keyi1 of lena image								Keyi2 of lena image							

and "keyi2" are combined with an external key to generate substitution-diffusion sequences, and consequently, substitution-diffusion sequences are also different for one image to another, and this makes the algorithm high secured and resistant against attacks.

2.1.2. Zigzag Process. The zigzag process aims to scramble the image by changing the position of the pixel or the bit. Depending on the context, it can act as a diffusion or substitution process. In this algorithm, the zigzag process applied to the whole binary image acts as a diffusion process. In the case of the binary subblock, it acts as a substitution process. A zigzag process to scramble the binary images is shown in Figure 2.

2.1.3. Affine Transformation Process. On the Euclidean plane, let w: $R^2 \longrightarrow R^2$ be an affine transformation; its equation is given by

$$\begin{aligned} w(x,y) &= (ax+by+e, cx+dy+f) \\ &= (x', y'), \end{aligned} \tag{1}$$

where (x, y) is the coordinate point, a, b, c, d, e, and f are real numbers, and (x', y') is the new coordinate point. We can also write this same transformation with the equivalent notations:

$$\begin{aligned} w(u) &= w\begin{pmatrix} x \\ y \end{pmatrix} \\ &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\ &= \begin{pmatrix} x' \\ y' \end{pmatrix} \\ &= Au + T, \end{aligned} \tag{2}$$

where A is a 2×2 real matrix and $T = \begin{pmatrix} e \\ f \end{pmatrix}$ represents translations.

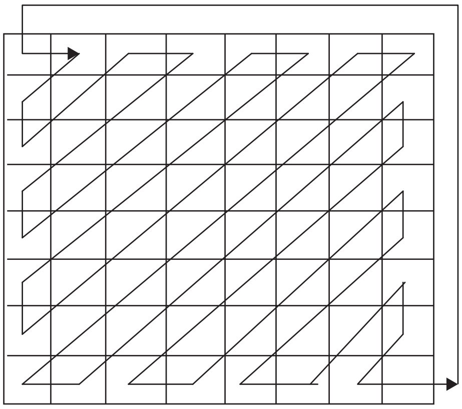


Figure 2: A zigzag path to scramble the binary subblock.

The matrix A can also be written in the form of

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 \cos \theta_1 & -r_2 \sin \theta_2 \\ r_1 \sin \theta_1 & r_2 \cos \theta_2 \end{pmatrix}, \tag{3}$$

where (r_1, θ_1) are the polar coordinates of the point (a, c) and $(r_2, (\theta_2 + (\pi/2)))$ are the polar coordinates of the point (b, d). In other words,

$$r_1 = \sqrt{a^2 + c^2}, \quad \tan \theta_1 = \frac{c}{a}, \qquad r_2 = \sqrt{b^2 + d^2}, \quad \tan \theta_2 = \frac{b}{d}. \tag{4}$$

Various transformations can be performed in R^2 such as dilations, reflections, translations, rotations, and similitudes. In this work, we are restricted to reflection and rotation transformations.

A reflection on the x-axis can be written in as $w_{rx}(x, y) = (x, -y)$, while a reflection on the y-axis is written as $w_{ry}(x, y) = (-x, y)$. In matrix representation, these reflections are given by

$$(x', y'),$$

$$w_{ry}(u) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{5}$$

A rotation mapping has the form $w_r(x, y) = (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta)$.

Also, it is expressed as

$$w_r(u) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{6}$$

For the rotation angle θ, $0 \leq \theta < 2\pi$.

The available reflection and rotation transformations are

(0) Identity (I):

(1) Orthogonal reflection around midvertical axis of block (Rmv)

(2) Orthogonal reflection around midhorizontal axis of block (Rmh)

(3) Orthogonal reflection around the first diagonal of block (Rfdiag)

(4) Orthogonal reflection around the second diagonal of block (Rsdiag)

(5) Rotation around the center of block +90° $(R + 90)$

(6) Rotation around the center of block +180°$(R + 180)$

(7) Rotation around the center of block −90°$(R\text{−}90)$

In the algorithm step, we use zigzag binary matrices named $Ibzi_{mxn}$, $i = 1, 2, \ldots, 8$. An external key is used to choose the corresponding type of transformation to be applied on each zigzag binary image $Ibzi_{mxn}$. Let «**ABCDEFGHIJKLMNOPRSTUVW*αβγηθλξρτφ***» be an external key; to choose the corresponding affine transformations, we extract eight numbers «**D, H, L, P, U, *β*, *λ*, *φ***» from an external key. Each number is converted into its corresponding octave number. The obtained octave number gives the corresponding transformation (from 0 to 7 as describe previously) to be applied on the corresponding zigzag binary image. An external key should be chosen such as all these obtained octave number must not be null. At the end of the affine transformation process, the transformed binary images are used to reconstruct a new gray-scale image I'_{mxn}. This image I'_{mxn} is subdivided into nonoverlapping squared block before the substitution-diffusion process.

2.1.4. Substitution Process. The operation is taken into two steps, zigzag operation and substitution, using the corresponding key. The chosen subblock SI'_{kxk} is converted into a binary subblock SbI'. The zigzag operation is taken on SbI', and it is reconverted into the pixel value. A zigzag path to scramble the binary block is shown in Figure 2. The next step is to combine an external key with «*keyi1*» to generate the sequence which will be used for substitution operation. Let L be the size of a subblock. Having an external key «*externalkey*» and an internal key «*substitutioncode*», the algorithm to generate the sequence of each block is defined as follows:

(a) For the first round and the first subblock,

```
Lon = length (keyi1);
lng = Lon/2;
codesub = keyi1;
cc1 = bitxor (externalkey, codesubu (1 : lng));
cc2 = bitxor (cc1, codesub ((lng + 1) : Lon));
usingcode = [cc1 cc2];
```

(b) For the next round K and the other subblocks,

```
codesubu = usingcode;
c1 = bitxor (clefdecfin, codesubu ((lng + 1) : Lon));
cc1 = bitxor (c1, vectsub (1 : 1 : lng));
c2 = bitxor (cc1, codesubu (lng:–1 : 1));
cc2 = bitxor (c2, vectsub ((lng + 1) : Lon));
usingcode = [cc1 cc2];
```

« vectsub » is the last result.

The sequence «usigncode» will be used for the substitution process of the corresponding subblock. The second step of the substitution process is performed using «XOR» operation.

2.1.5. Diffusion Process. The subblock coming from the substitution process is then sent through the permutation process. This process will modify the pixel location. The same algorithm used during the substitution process is used to generate the sequence for the diffusion process in which « keyi2» is used. The sequence of each subblock is rearranged in the ascending order and used to modify the position of the pixel in the subblock.

2.1.6. The Overall Proposed Encryption Algorithm. Let I_{mxn} be an original image. We can describe our encryption algorithm as follows:

```
Step 1: generate an external secret key.
Step 2: convert image $Ibi_{mxn}$ images.
Step 3: extract an internal secret key.
Step 4: apply zigzag process on binary images.
Step 5: select and apply affine transformation on each zigzag binary images using the external key.
Step 6: reconstruct a new gray-scale image using transformed zigzag binary images.
Step 7: subdivide a new image into many nonoverlapping squared subblock.
Step 8: for each subblock and for $K$ round,
    Step 8.1: convert each pixel into the binary vector.
    Step 8.2: apply scan zigzag operation to permute the position of the binary element in the binary matrix.
    Step 8.3: reconstruct a gray-scale subblock.
    Step 8.4: combine an external key and an internal key to generate the substitution sequence of the subblock.
    Step 8.5: take out a second part of the substitution process.
    Step 8.6: combine an external key and an internal key to generate the permutation sequence of the subblock.
    Step 8.7: take out a diffusion process.
```

The decryption algorithm is the inverse operations of the encryption process.

2.2. Evaluation Metrics. A robust and good cryptosystem should present many features. Firstly, the key space should be large enough to make the brute-force attack infeasible [17]. Secondly, the histograms of the cipher image should be uniformly distributed. The correlation between adjacent pixels (vertically, horizontally, and diagonally) in the cipher image should be approximately zero to confirm the effectiveness of the method.

2.2.1. Correlation Coefficient. The correlation metric is used to evaluate the similarity between two images. If the images are identical, the correlation value is equal to one. When the correlation value is closed to zero, there is no similarity between these images. For an efficient encryption scheme, the correlation between plain image and cipher image must be close to zero. The correlation coefficient (Co) between original and encrypted images is defined as follows:

$$Co = \frac{N_p \sum_{j=1}^{N_p}\left(x_j \times y_j\right) - \sum_{j=1}^{N_p} x_j \times \sum_{j=1}^{N_p} y_j}{\sqrt{N_p \sum_{j=1}^{N_p} x_j^2 - \left(\sum_{j=1}^{N_p} {x_j}^2\right) \times \left(N_p \sum_{j=1}^{N_p} y_j^2 - \left(\sum_{j=1}^{N_p} y_j\right)^2\right)}}, \tag{7}$$

where x and y are gray-scale pixel values of the original and encrypted images and N_p is the total number of pixels.

The correlation coefficient γ of each pair of adjacent pixels is calculated using [42]

$$\gamma(x, y) = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)\sqrt{D(y)}}}, \tag{8}$$

where

$$\operatorname{cov}(x, y) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} \left[x_i - E(x)\right]\left[y_i - E(y)\right], \tag{9}$$

$$E(x) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} x_i, \tag{10}$$

$$D(x) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} \left[x_i - E(x)\right]^2. \tag{11}$$

In equations (8)–(11), x and y are the gray values of two adjacent pixels in the image and N_{ap} is the total number of adjacent pairs of pixels. $E(x)$ is the expectation of variable x, $D(x)$ is the variance, and $\operatorname{cov}(x, y)$ is the covariance of two adjacent pixels in the image. For a good cryptosystem, the correlation coefficient Co between plain and cipher images should be close to zero. Likewise, the correlation coefficient γ of each pair of adjacent pixels in the cipher image should be close to zero.

2.2.2. Entropy Information. The information entropy, introduced by Shannon, is one of the most important features of randomness. It is used to evaluate the quantity of information in the image. Information entropy $H(s)$ is calculated in [28] using

$$H(s) = \sum_{i=0}^{N_{gl}-1} P(s_i)\log_2\left(\frac{1}{p(s_i)}\right), \tag{12}$$

where N_{gl} is the total number of gray levels in the image and $P(s_i)$ shows the probability of appearance of the symbol s_i. The entropy value of encrypted images should be closed to eight.

The (k, TB)-local Shannon entropy with respect to local image blocks may be computed by the following steps [43]. First, nonoverlapping image blocks S1, S2, . . . Sk with TB pixels for a test image S are randomly selected. Then, information entropy $H(S_i)$ for all image blocks via equation (12) may be obtained. Finally, the local Shannon entropy over these k image blocks is computed using

$$H_{k,T_B}(m) = \sum_{i=1}^{k} \frac{H(S_i)}{k}. \quad (13)$$

2.2.3. Differential Attacks. The change of a single pixel on a plain image should have an important effect on the cipher image. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are used to test the influence of changing a single pixel in the original image on the whole encrypted image [17]. Therefore, if $A(i, j)$ and $B(i, j)$ are the pixels in row i and column j of the encrypted images A and B, with only one-pixel difference between the respective plain images, then the NPCR is calculated by using the following formula [44]:

$$\text{NPCR} = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\%, \quad (14)$$

where W and H are the width and height of A or B. $D(i, j)$ is calculated as follows:

$$D(i, j) = \begin{cases} 1, & \text{if } A(i, j) \neq B(i, j), \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

UACI is calculated by the following formula:

$$\text{UACI}(A, B) = \frac{1}{W \times H} \left[\sum_{i,j} \frac{|A(i, j) - B(i, j)|}{255} \right] \times 100\%. \quad (16)$$

The estimated score for NPCR and UACI are 99.6094% and 33.4635% for gray-scale images [45].

2.2.4. Image Quality Criterion. After the encryption/decryption processes, we need to evaluate the performance of the cryptosystem and the quality of images. This is done by evaluating the mean square error (MSE), the peak signal-to-noise ratio, (PSNR) and the encryption quality.

Let P and P', respectively, be a plain image, an encrypted image, and a decrypted image. MSE is defined as follows:

$$\text{MSE} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} \left[P(m, n) - P'(m, n) \right]^2}{M \times N}, \quad (17)$$

where M is the total number of lines in the image and N is the total number of columns. The PSNR is defined as follows:

$$\text{PSNR} = 10\text{Log}_{10}\left(\frac{255^2}{\text{MSE}} \right). \quad (18)$$

When the decrypted image P' is identical to the plain one, MSE is zero, and consequently, PSNR is infinite.

The total changes in pixels' values between the plain image and the encrypted image allow to confirm the encryption quality. So, encryption quality gives us the average number of changes to each gray level between plain image and its corresponding encrypted image. It is defined as follows:

$$\overrightarrow{\text{Encryption quality}} = \frac{\sum_{L}^{255} \left| H_L(C) - H_L(P) \right|}{256}, \quad (19)$$

where C denotes the encrypted image, L is the gray level, $L = 0, 1, 2, \ldots 255$, and H_L is the total number of occurrence of L in the image.

3. Experimental Results

In this work, we used medical and nonmedical images from different databases. Some of the medical images (parasite images) have been taken from different hospital laboratories. The others are from [46, 47]. Nonmedical images are from [48]. Azafack and Guefack images have been taken with a Smartphone techno-Y4. Our algorithm should be carried out using MATLAB R2014a in COMPAQ Intel ® core ™ i3-2328M CPU @ 2.20 GHz 2.20 GHz. The time for encrypting/decrypting an image of 512 × 512 is 1.1 s. The times for encrypting/decrypting of many images are presented in Table 2.

3.1. Visual Test. To appreciate the effect of the zigzag and affine transformation processes, we present in Figure 3 some plain images and their corresponding transformed images. Physically, these processes have destroyed the correlations between the adjacent pixels in the plain images. Figure 4 presents some encrypted and decrypted images using our proposed cryptosystem where the external key is "A23C56789ABADEF7167DEAB6789367A9». The size of the subblock is 4 × 4 pixels and the number of rounds on one subblock is five. It is obvious from visual inspection of Figures 4(a)–4(h)) that there is no correlation between the original image and encrypted image. It is therefore impossible, by observing the encrypted image, to deduce the original image. This ensures the physical privacy of the cryptosystem. When observing original and decrypted images in Figures 4(a), 4(c), 4(d), 4(f), 4(g), and 4(i)), it is obvious that the image decrypted is similar to the original image. This test was performed on many other images using different keys, and all the results were conclusive. Thus, visually, the efficiency of the cryptosystem is guaranteed. Thereafter, we conducted a statistical analysis in order to confirm the results of the visual tests.

3.2. Key space Analysis. The key space of a good image encryption algorithm should be large enough to make any brute-force attack ineffective [17, 42]. The proposed algorithm used two keys. Firstly, an external key of 128 bits is long; thus, the cipher image has 2^{128} different combinations of the secret key. Secondly, we use an internal secret key of 32 or 64 gray values which are long coming from the decomposition of an original image. So, the size of the key is large enough.

3.3. Correlation Tests

3.3.1. Correlation between Plain and Cipher Image. In Table 3, the correlation coefficients between plain and cipher images of several medical gray-scale images are given. It is

Table 2: Encryption/decryption time.

Image size	Encryption/decryption time (second)
512 × 512	1.1
256 × 256	0.553
128 × 128	0.289
64 × 64	0.104

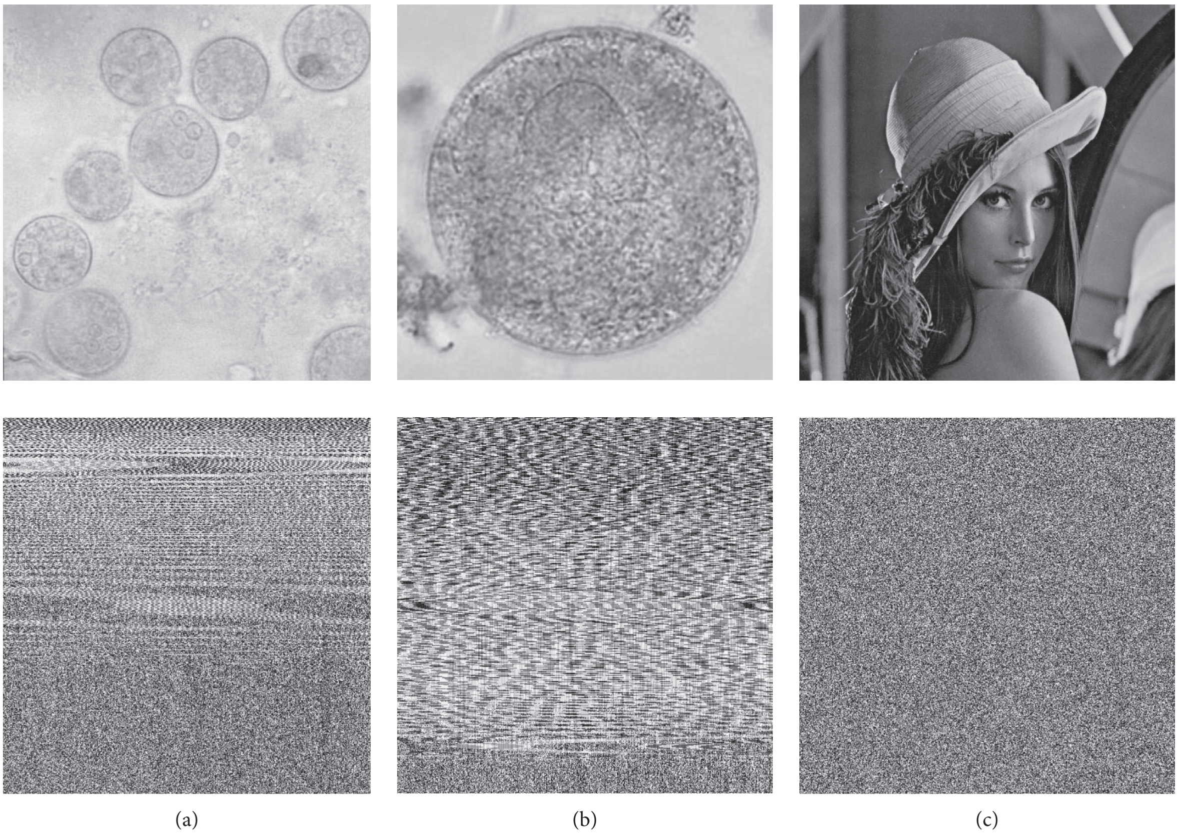
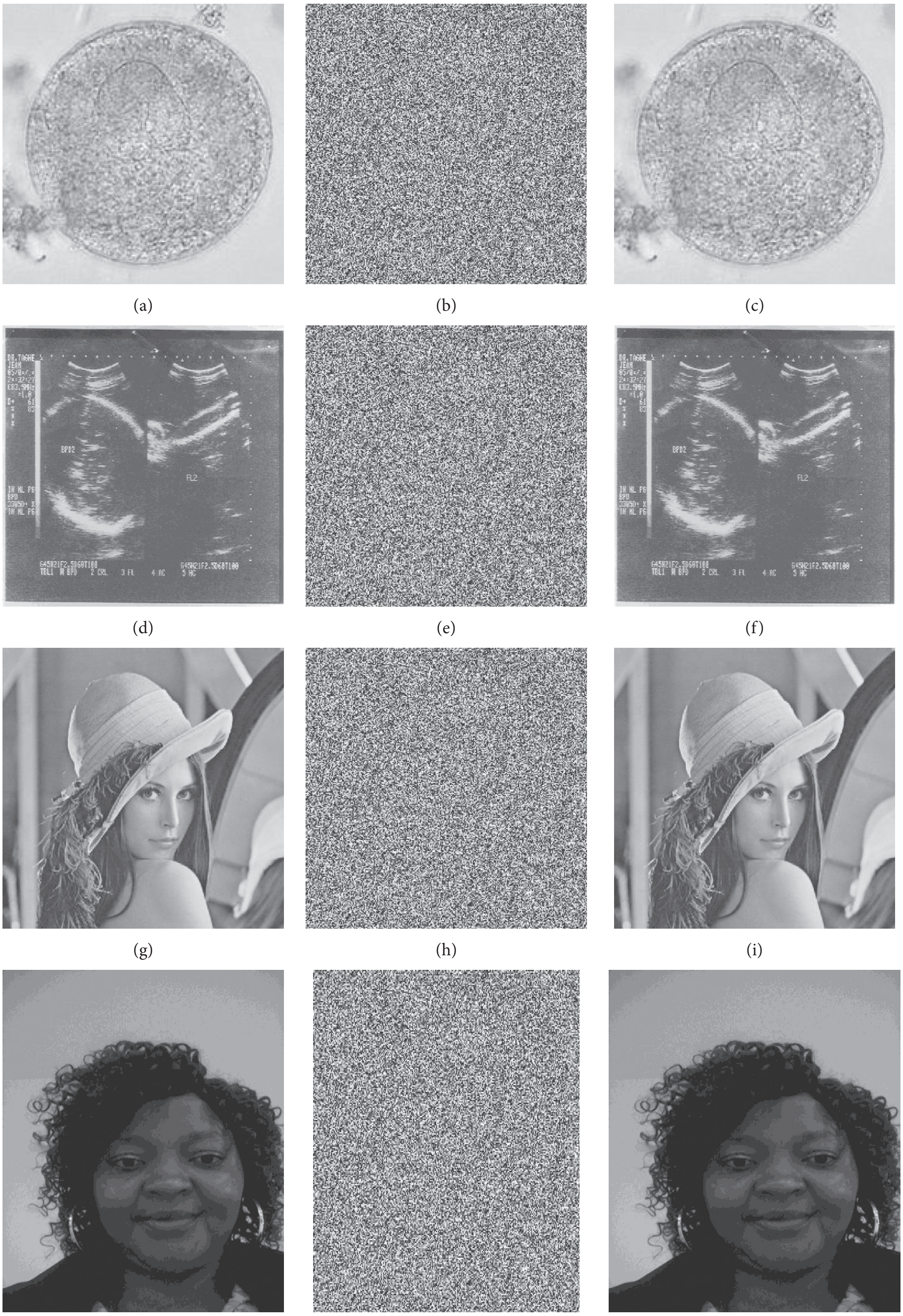


Figure 3: Effect of zigzag and affine transformation processes on some images using the external key «A23C56789AB-ADEF7167DEAB6789367A9»: (a) "Antamoebacoli" and its corresponding transformed image. (b) A plain image "Balantidum Coli cyst" and its corresponding transformed image. (c) A plain image "Girl (Lena, 4.2.04)" and its corresponding transformed image.

observed that all the correlation coefficients are negligible. The highest value (0, 00348) is obtained for the "*ossify*" image. The cipher images are not correlated with plain images. The correlation between the decrypted and the original images is always "1" confirming that both images are identical. The algorithm has been applied on other types of nonmedical images. We have used the USC-SIPI image database which is a collection of the digitized image available and maintained by the University of Southern California [42]. Table 4 shows the results of those images. The same as for medical images, correlation coefficients of images in Table 4 are closed to zero. The maximum value (−0, 00504) of the correlation coefficient is very low compared to the critical value (1). This confirms that the proposed algorithm is efficient for every type of image.

3.3.2. Correlation of Adjacent Pixels. Table 5 shows the correlation coefficients between two vertically, two horizontally, and two diagonally adjacent pixels in several medical plain images and also in their corresponding encrypted images. A high correlation is noted between vertically, horizontally, and diagonally adjacent pixels of original images. The lower value (0.67636) is obtained between diagonal adjacent pixels on the image «ANTAMOEBACOLI». For encrypted images, these values are approximately zero showing that two vertically, two horizontally, and two diagonally adjacent pixels of encrypted images are not correlated. The highest value (−0, 00514) is obtained in the case of medical images «article*_oeuf_taeniaC2*». This is a significant feature proving the effectiveness of our cryptosystem. The same observation is made in the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4: Visual test on some images using the secret key «A23C56789ABADEF7167DEAB6789367A9»: (a), (b), and (c) A plain image "Balantidum Coli cyst" and its corresponding cipher and decrypted image, respectively. (d), (e), and (f) A plain image "echopelv" and its corresponding cipher and decrypted image, respectively. (g), (h), and (i) A plain image "Girl (Lena, 4.2.04)" and its corresponding cipher and decrypted image, respectively. (j), (k), and (l) A plain image "Guefack" and its corresponding cipher and decrypted image, respectively.

Table 3: Correlation coefficients and entropy information of some medical images.

Image name	Size	Correlation coefficients between plain and cipher images	Entropy value of plain image	Entropy value of cipher image	Correlation coefficients between plain and decrypted images
ANTAMOEBACOLI	398 × 407	−1.27*e*−03	6.9714	7.9993	1
article_oeuf_taeniaC2	200 × 200	3.23 *e*−03	7.2348	7.9993	1
Balantidium coli cyst	200 × 200	−2.54*e*−03	7.0118	7.9993	1
Balantidium coli_trophozoite	200 × 200	−1.76*e*−03	5.6742	7.9992	1
DICROCOELIUM	400 × 341	1.66*e*−03	6.6175	7.9994	1
Entamoeba coli trophozoite	200 × 200	−9.34*e*−04	7.1234	7.9992	1
Entamoeba histolytica cyst	200 × 200	−1.08*e*−03	5.8102	7.9994	1
Entamoeba histolytica-cyst-Gini	130 × 130	1.58*e*−03	6.5016	7.9993	1
Entamoeba histolytica trophozoite	200 × 200	8.49*e*−04	7.6217	7.9994	1
Entamoeba histolytica trophozoite_redim	120 × 120	−3.26*e*−03	7.8522	7.9992	1
Entamoeba histolytica trophozoite_redim2	172 × 160	7.73*e*−05	7.7367	7.9993	1
oeuf_ascarisc	266 × 200	−1.48*e*−04	6.9107	7.9993	1
S- Hematobium egg	400 × 300	−1.03*e*−03	7.1983	7.9992	1
S- Mansoni egg	400 × 300	9.58*e*−04	6.4561	7.9993	1
tropho_entamoeba_histolytica2	332 × 213	−1.99*e*−03	3.8234	7.9993	1
tropho_iodamoeba_butschlii	200 × 200	7.70*e*−04	6.3247	7.9993	1
Angio	64 × 64	−3.21*e*−03	7.2769	7.9992	1
DisLocElbow	64 × 64	9.58*e*−04	5.5326	7.9993	1
echo1	64 × 64	1.61*e*−03	6.3281	7.9993	1
I1_200	64 × 64	−7.02*e*−04	6.4746	7.9993	1
Node2	64 × 64	−8.81*e*−04	6.8732	7.9994	1
Ossify	64 × 64	3.48*e*−03	6.9989	7.9993	1
Pelvis	64 × 64	1.29*e*−03	6.4653	7.9993	1
Ribs	64 × 64	1.12*e*−03	6.2298	7.9991	1
Dirofilaria	241 × 500	−2.59*e*−03	6.7666	7.9993	1
Headirm	256 × 256	1.38*e*−03	5.0299	7.9993	1
Abdomenirm	256 × 256	−6.18*e*−04	6.9551	7.9993	1
Pelvisirm	256 × 256	1.08*e*−03	6.7379	7.9993	1
Gastrointestinal_parasites	512 × 392	5.58*e*−07	7.6634	7.9994	1
Echo fetus at 12 weeks	300 × 210	3.41*e*−04	6.4062	7.9993	1
Ultrasound of fetus of 3months	512 × 396	1.90*e*−03	7.2999	7.9994	1
Echopelv	601 × 711	−6.74*e*−04	6.5896	7.9998	1
CT-MONO2-8-abdo	128 × 128	−4.3347*e*−04	5.8925	7.9992	1
OT-MONO2-8-colon	128 × 128	−2.7*e*−03	6.7468	7.9993	1

case of nonmedical images in Table 6. The proposed cryptosystem produces encrypted images completely different from original images. In Figure 5, we have presented the distribution of horizontally and vertically adjacent pixels in the plain image and its corresponding encrypted image.

3.4. Histograms. The histograms of both the plain and the encrypted images using our proposed method are shown in Figure 6. The histograms of the encrypted images are uniformly distributed while those of plain images are not. This confirms the toughness of the method over any statistical attack. The security of the encryption method is therefore very strong.

3.5. Key Sensitivity Test. An ideal cipher image should be extremely sensitive with respect to the key used in the algorithm during the encryption and decryption processes.

3.5.1. Key Sensitivity Test of Encryption Process. An insignificant change in the encryption key should be sensitive to the cipher images. We take out the key sensitivity test during the encryption process by using different keys to encrypt the same image. The difference between keys is a single bit change. The procedure is described as follows:

(a) We used «A23C56789ABADEF7167DEAB67893 67A9» as first external key to encrypt«S-Hematobium egg» image. The encrypted image is presented on Figure 7(b).

(b) We change a single bit on the external key by substituting the fourth character "C" into "D." The external key in this case becomes «A23D56789ABADEF7167DEAB6789367A9». The obtained encrypted image is presented on Figure 7(c).

(c) In the second case, we change the sixteenth character 7 in the external key into 6. The used external key is

Table 4: Correlation coefficients and entropy information of some nonmedical images.

File name	Description	Size	Correlation coefficients between plain and cipher images	Entropy value of plain image	Entropy value of cipher image	Correlation coefficients between plain and decrypted images
4.1.01	Girl	256 × 256	3.46e−03	7.1835	7.9994	1
4.1.02	Couple	256 × 256	1.05e−03	6.5007	7.9993	1
4.1.03	Girl	256 × 256	6.15e−04	5.9549	7.9994	1
4.1.04	Girl	256 × 256	−2.82e−04	7.6031	7.9993	1
4.1.05	House	256 × 256	1.95e−03	7.0902	7.9993	1
4.1.06	Tree	256 × 256	−3.07e−03	7.5634	7.9993	1
4.1.07	Jelly beans	256 × 256	−1.85e−03	6.6098	7.9993	1
4.1.08	Jelly beans	256 × 256	1.44e−03	6.8863	7.9993	1
4.2.01	Splash	512 × 512	9.15e−04	7.3232	7.9994	1
4.2.02	Girl (tiffany)	512 × 512	−2.28e−03	6.6691	7.9993	1
4.2.03	Mandrill (a.k.a. Baboon)	512 × 512	−5.77e−04	7.7659	7.9993	1
4.2.04	Girl (lena. or lena)	512 × 512	4.40e−04	7.7548	7.9993	1
4.2.05	Airplane (F-16)	512 × 512	1.89e−04	6.6879	7.9994	1
4.2.06	Sailboat on lake	512 × 512	5.04e−03	7.7675	7.9994	1
4.2.07	Peppers	512 × 512	2.20e−03	7.7253	7.9993	1
5.1.09	Moon surface	256 × 256	−1.16e−03	6.719	7.9993	1
5.1.10	Aerial	256 × 256	1.71e−03	7.322	7.9992	1
5.1.11	Airplane	256 × 256	−3.61e−03	6.4658	7.9994	1
5.1.12	Clock	256 × 256	1.95e−03	6.7111	7.9993	1
5.1.13	Resolution chart	256 × 256	−1.47e−03	2.2863	7.9994	1
5.1.14	Chemical plant	256 × 256	8.36e−05	7.3473	7.9993	1
5.2.08	Couple	512 × 512	−5.45e−04	7.2187	7.9994	1
5.2.09	Aerial	512 × 512	1.37e−03	7.0015	7.9994	1
5.2.10	Stream and bridge	512 × 512	−1.92e−03	7.721	7.9992	1
7.1.01	Truck	512 × 512	8.68e−04	6.5836	7.9993	1
7.1.02	Airplane	512 × 512	−3.11e−05	5.4408	7.9993	1
7.1.03	Tank	512 × 512	−1.54e−03	6.4078	7.9994	1
7.1.04	Car and APCs	512 × 512	−1.91e−04	6.8064	7.9992	1
7.1.05	Truck and APCs	512 × 512	−1.27e−03	7.1124	7.9994	1
7.1.06	Truck and APCs	512 × 512	2.49e−03	7.0571	7.9992	1
7.1.07	Tank	512 × 512	2.88e−04	6.5399	7.9993	1
7.1.08	APC	512 × 512	−2.56e−03	5.9022	7.9993	1
7.1.09	Tank	512 × 512	1.01e−03	6.9868	7.9993	1
7.1.10	Car and APCs	512 × 512	−1.11e−03	6.6201	7.9994	1
boat.512	Fishing boat	512 × 512	9.47e−04	7.2151	7.9992	1
elaine.512	Girl (elaine)	512 × 512	5.99e−04	7.5118	7.9992	1
House	House	512 × 512	1.67e−03	6.5802	7.9993	1
gray21.512	21 level step wedge	512 × 512	1.54e−03	4.5922	7.9992	1
numbers. 512	256 level test pattern	512 × 512	−1.56e−03	7.7768	7.9994	1
	Azafack	398 × 512	−1.19e−03	7.7018	7.9994	1
	Guefack	365 × 486	8.64e−04	6.6996	7.9994	1

then «A23C56789ABADEF6167DEAB6789367A9». The encrypted image is shown in Figure 7(d).

(d) In the test number 3, the used external key is changed into «A23C56789ABADEF7167DEAB6789367AA». Here, the last character 9 in the first external key is changed into A. Figure 7(e) presents a cipher image obtained.

(e) For the last test, the second character 2 in the external key is changed into 3. The used external key becomes «A33C56789ABADEF7167DEAB6789367A9». Figure 7(f) presents a cipher image obtained.

For a quantitative assessment of the similarity between those images, we present in Table 7 the correlation coefficients between the different cipher images.

Although the difference from one key to the other is a single bit, we note from Table 7 that the values of the correlations' coefficients between the different encrypted images obtained are closed to zero. The highest correlation

Table 5: Correlation coefficients between two vertically, horizontally, and diagonally adjacent pixels in several medical plain images and also in their corresponding encrypted images.

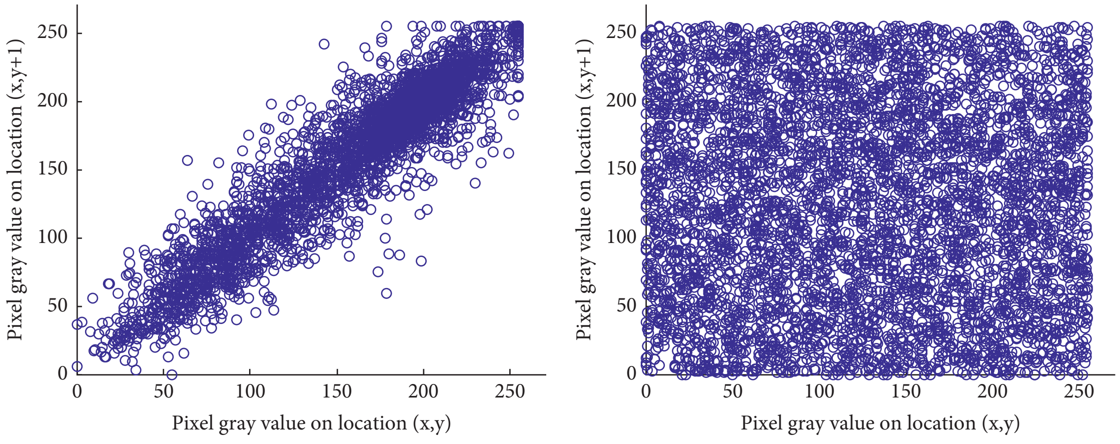
Image name	Correlation coefficients of original images			Correlation coefficients of cipher images		
	Vert. cor	Hor. cor	Diag. cor	Vert. cor	Hor. cor	Diag. cor
ANTAMOEBACOLI	0.77141	0.81532	0.67636	−2.96*e*−03	1.31*e*−03	1.06*e*−05
Article_oeuf_taeniaC2	0.93107	0.9308	0.87861	−1.94*e*−03	−5.14*e*−03	7.55*e*−04
Balantidium coli cyst	0.76953	0.78613	0.66231	1.74*e*−03	−6.15*e*−04	6.42*e*−04
Balantidium coli_trophozoite	0.9259	0.93259	0.90567	2.19*e*−03	−3.48*e*−03	1.56*e*−04
DICROCOELIUM	0.97729	0.98048	0.96751	3.89*e*−04	3.14*e*−03	−5.42*e*−04
Entamoeba coli trophozoite	0.99129	0.99215	0.98588	−1.85*e*−03	−9.26*e*−04	3.17*e*−03
Entamoeba histolytica cyst	0.93725	0.94537	0.90185	2.47*e*−03	−2.75*e*−03	−2.22*e*−03
Entamoeba histolytica-cyst-Gini	0.92637	0.91525	0.85601	1.69*e*−03	−2.96*e*−03	−2.11*e*−04
Entamoeba histolytica trophozoite	0.98471	0.98541	0.97413	2.36*e*−03	2.87*e*−03	9.35*e*−04
Entamoeba histolytica trophozoite_redim	0.98253	0.98307	0.96996	−1.73*e*−03	−2.91*e*−03	−3.19*e*−04
Entamoeba histolytica trophozoite_redim2	0.98622	0.9867	0.97545	2.20*e*−03	1.51*e*−04	−8.70*e*−05
oeuf_ascarisc	0.93155	0.93445	0.88098	2.59*e*−04	−1.48*e*−03	−6.88*e*−04
S-Hematobium egg	0.88554	0.92383	0.8354	−1.64*e*−03	−2.70*e*−03	−1.60*e*−03
S-Mansoni egg	0.86917	0.9053	0.80198	−3.42*e*−03	4.26*e*−04	1.10*e*−03
Tropho_entamoeba_histolytica2	0.99265	0.99139	0.98605	3.29*e*−03	5.36*e*−05	1.42*e*−03
Tropho_iodamoeba_butschlii	0.9966	0.99688	0.99391	−4.73*e*−04	5.99*e*−04	−1.66*e*−03
Angio	0.89905	0.96795	0.90462	−1.78*e*−03	3.27*e*−03	−8.62*e*−04
DisLocElbow	0.96446	0.99666	0.9608	1.40*e*−03	5.22*e*−04	1.54*e*−03
Echo1	0.88904	0.8776	0.8212	1.77*e*−03	−2.03*e*−04	3.88*e*−04
I1_200	0.99393	0.98775	0.98353	7.66*e*−04	8.84*e*−04	1.20*e*−03
Node2	0.96491	0.95401	0.93398	3.21*e*−03	2.26*e*−03	−9.21*e*−04
Ossify	0.99017	0.94411	0.9303	2.44*e*−03	2.14*e*−03	9.73*e*−04
Pelvis	0.94299	0.99754	0.93903	−1.73*e*−03	1.78*e*−03	−3.68*e*−03
Ribs	0.87322	0.88588	0.85227	−2.05*e*−04	−8.69*e*−04	1.70*e*−04
Dirofilaria	0.98814	0.97363	0.96349	−9.60*e*−04	−8.31*e*−04	−4.47*e*−04
Headirm	0.96343	0.95734	0.93986	−2.08*e*−03	6.53*e*−04	1.32*e*−03
Abdomenirm	0.91929	0.91331	0.8508	−1.35*e*−03	−6.20*e*−05	3.75*e*−04
Pelvisirm	0.96602	0.95567	0.93225	−9.36*e*−04	4.49*e*−04	2.93*e*−04
Gastrointestinal_parasites	0.91404	0.94614	0.88418	−3.06*e*−04	−2.40*e*−03	5.55*e*−05
Echo fetus at 12 weeks	0.91943	0.90556	0.85116	7.24*e*−04	−3.15*e*−03	−1.24*e*−03
Ultrasound of fetus of 3 months	0.96781	0.98601	0.95989	−1.85*e*−03	−1.60*e*−03	2.34*e*−03
Echopelv	0.89428	0.90612	0.82813	−9.96*e*−04	−1.11*e*−03	3.76*e*−04
CT-MONO2-8-abdo	0.93917	0.95915	0.91275	−2.57*e*−03	−3.56*e*−03	−2.56*e*−04
OT-MONO2-8-colon	0.96189	0.96556	0.93819	−2.33*e*−03	5.95*e*−04	−4.25*e*−04

Table 6: Correlation coefficients between two vertically, horizontally, and diagonally adjacent pixels in several nonmedical plain images and also in their corresponding encrypted images from database.

Image Name	Description	Correlation coefficients of original images			Correlation coefficients of cipher images		
		Vert cor	Hor cor	Diag cor	Vert cor	Hor cor	Diag cor
4.1.01	Girl	0.96547	0.9737	0.94928	-2.94*e*−04	1.58*e*−03	5.41*e*−04
4.1.02	Couple	0.95615	0.93889	0.90658	2.09*e*−03	-1.85*e*−04	9.21*e*−04
4.1.03	Girl	0.91432	0.97598	0.89425	-4.05*e*−04	2.47*e*−03	4.50*e*−04
4.1.04	Girl	0.98476	0.96995	0.95805	5.88*e*−05	5.11*e*−04	-1.98*e*−03
4.1.05	House	0.95289	0.9781	0.94157	-3.75*e*−04	-1.20*e*−03	2.38*e*−04
4.1.06	Tree	0.9441	0.96695	0.9285	-2.87*e*−03	-3.22*e*−03	5.65*e*−04
4.1.07	Jelly beans	0.98233	0.9787	0.96461	1.58*e*−03	-4.40*e*−03	-2.56*e*−03
4.1.08	Jelly beans	0.97553	0.97258	0.95246	3.56*e*−03	-7.71*e*−04	-6.16*e*−04
4.2.01	Splash	0.9915	0.98399	0.98054	2.25*e*−04	-1.19*e*−03	-8.85*e*−04
4.2.02	Girl (tiffany)	0.94097	0.93826	0.91514	-2.56*e*−03	-8.88*e*−04	1.63*e*−03
4.2.03	Mandrill (a.k.a. Baboon)	0.75486	0.86269	0.72324	1.36*e*−03	-5.84*e*−04	-4.73*e*−04
4.2.04	Girl (lena. or lena)	0.98485	0.97159	0.9635	9.09*e*−05	-2.57*e*−04	-4.52*e*−04
4.2.05	Airplane (F-16)	0.96394	0.96616	0.94106	-7.67*e*−04	2.28*e*−03	-9.57*e*−05
4.2.06	Sailboat on lake	0.97003	0.97368	0.95768	9.21*e*−04	5.35*e*−04	3.48*e*−04

Table 6: Continued.

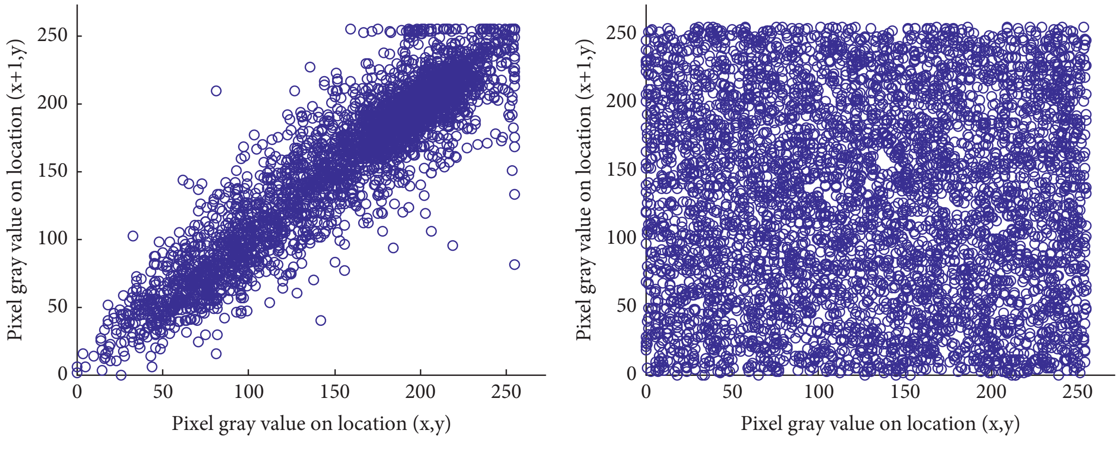
Image Name	Description	Correlation coefficients of original images			Correlation coefficients of cipher images		
		Vert cor	Hor cor	Diag cor	Vert cor	Hor cor	Diag cor
4.2.07	Peppers	0.97788	0.97567	0.96806	3.40*e*−03	-1.47*e*−03	-3.18*e*−04
5.1.09	Moon surface	0.93093	0.89444	0.88385	3.49*e*−03	2.34*e*−03	-1.91*e*−03
5.1.10	Aerial	0.85731	0.90234	0.80779	3.27*e*−04	-1.65*e*−03	1.42*e*−03
5.1.11	Airplane	0.93722	0.95697	0.91391	6.44*e*−04	-1.97*e*−03	-1.84*e*−03
5.1.12	Clock	0.97373	0.95613	0.93755	-1.33*e*−03	-1.34*e*−03	1.06*e*−03
5.1.13	Resolution chart	0.86678	0.87242	0.75713	1.78*e*−03	8.89*e*−04	-2.11*e*−04
5.1.14	Chemical plant	0.89647	0.94525	0.8672	8.69*e*−04	-3.28*e*−03	-2.29*e*−03
5.2.08	Couple	0.89244	0.93684	0.85528	1.14*e*−04	9.13*e*−04	1.12*e*−03
5.2.09	Aerial	0.85803	0.89863	0.80248	-1.94*e*−03	-2.45*e*−04	-6.92*e*−04
5.2.10	Stream and bridge	0.92548	0.93854	0.89787	1.79*e*−03	1.26*e*−03	8.91*e*−04
7.1.01	Truck	0.91738	0.95927	0.90241	-1.88*e*−03	4.08*e*−03	-1.90*e*−04
7.1.02	Airplane	0.94637	0.9467	0.91936	-1.35*e*−03	2.23*e*−03	1.04*e*−03
7.1.03	Tank	0.92665	0.94073	0.90446	-2.69*e*−03	-3.46*e*−03	-2.49*e*−03
7.1.04	Car and APCs	0.96614	0.97538	0.95392	-1.17*e*−03	6.70*e*−04	1.46*e*−03
7.1.05	Truck and APCs	0.90661	0.93685	0.88725	-4.51*e*−03	1.12*e*−03	-8.10*e*−04
7.1.06	Truck and APCs	0.90045	0.93465	0.88065	-3.92*e*−03	6.38*e*−04	3.34*e*−04
7.1.07	Tank	0.86813	0.8768	0.82074	-3.03*e*−03	-2.04*e*−04	-1.32*e*−03
7.1.08	APC	0.92423	0.95333	0.91346	-1.88*e*−03	3.75*e*−03	2.08*e*−04
7.1.09	Tank	0.92675	0.96239	0.91523	-3.11*e*−03	1.84*e*−03	-2.76*e*−03
7.1.10	Car and APCs	0.94504	0.96181	0.92815	-2.16*e*−04	-3.24*e*−03	7.66*e*−04
Boat.512	Fishing boat	0.96993	0.93638	0.92308	5.46*e*−04	-3.22*e*−03	1.83*e*−04
Elaine.512	Girl (elaine)	0.9697	0.97256	0.96796	-1.19*e*−03	5.92*e*−04	-1.12*e*−03
House	House	0.95059	0.97305	0.93846	4.29*e*−03	-8.68*e*−04	-2.47*e*−03
Gray21.512	21-level step wedge	0.99984	0.99653	0.99636	1.55*e*−03	-1.49*e*−03	8.19*e*−04
Numbers.512	256-level test pattern	0.71603	0.73889	0.64186	1.32*e*−03	-3.03*e*−05	1.43*e*−03
	Azafack	0.95872	0.94002	0.92429	1.94*e*−04	3.10*e*−03	-5.24*e*−04
	Guefack	0.99057	0.99022	0.98472	1.60*e*−03	5.02*e*−04	6.64*e*−04



(a)

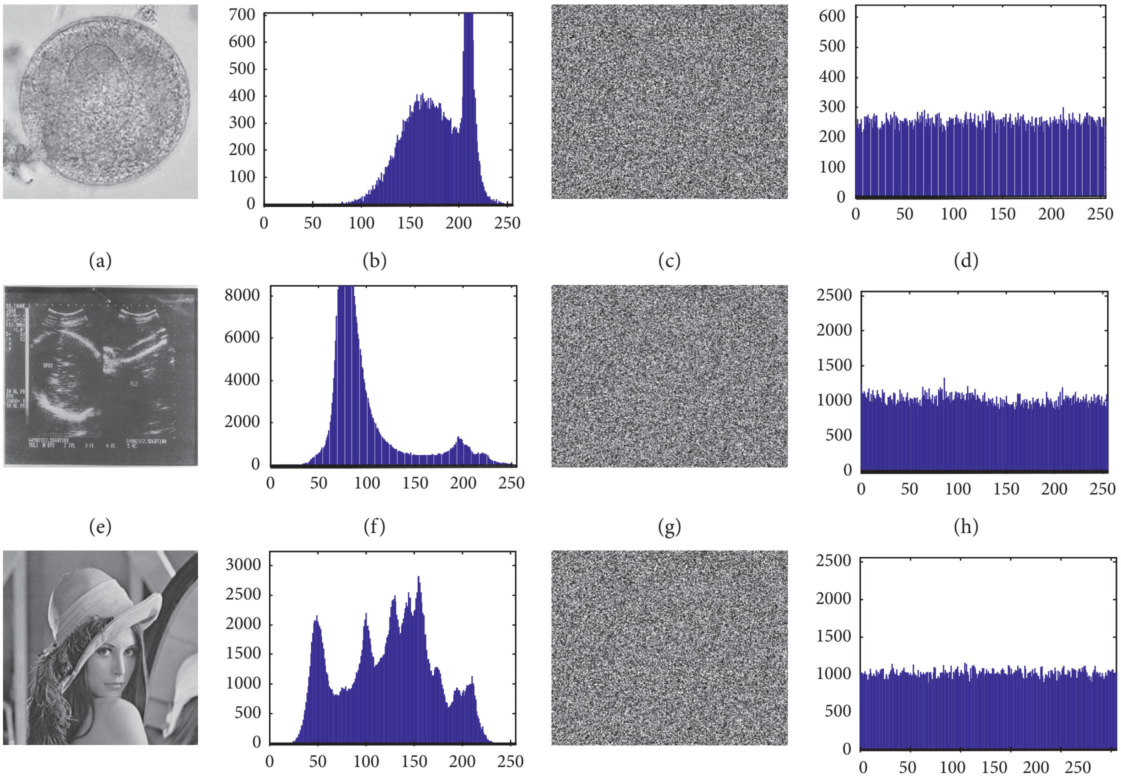
(b)

Figure 5: Continued.



(c) (d)

Figure 5: Correlation of adjacent pixels in «article_oeuf_taeniaC2.jpg image. (a) and (c) The distribution of horizontal and vertical adjacent pixels in the plain image. (b) and (d) The distribution of horizontal and vertical adjacent pixels in the corresponding encrypted image.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 6: Continued.

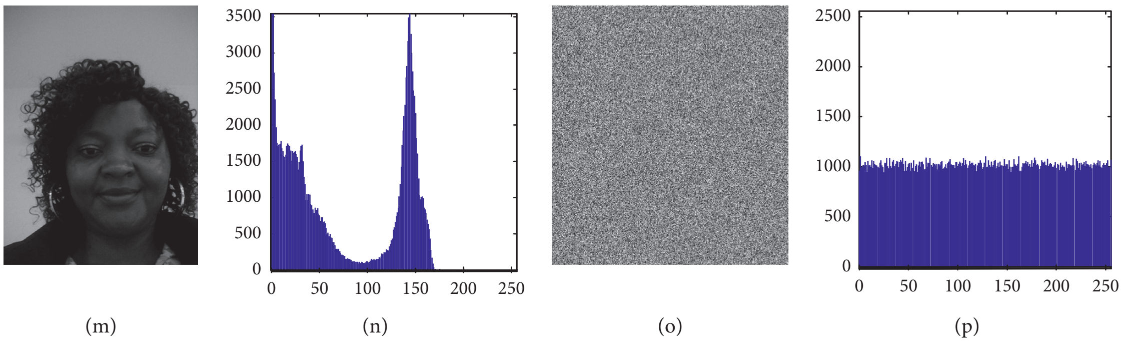


Figure 6: Histogram analysis plain and cipher images using the secret key «A23C56789ABADEF7167DEAB6789367A9»: (a), (b), (c), and (d) A plain image "Balantidium Coli cyst" and its corresponding histogram, cipher image, and cipher image histogram, respectively. (e), (f), (g), and (h) A plain image "echopelv» and its corresponding histogram, cipher image, and cipher image histogram, respectively. (i), (j), (k), and (l) A plain image "Girl (Lena, 4.2.04)» and its corresponding histogram, cipher image, and cipher image histogram, respectively. (m), (n), (o), and (p) A plain image "Guefack» and its corresponding histogram, cipher image, and cipher image histogram, respectively.

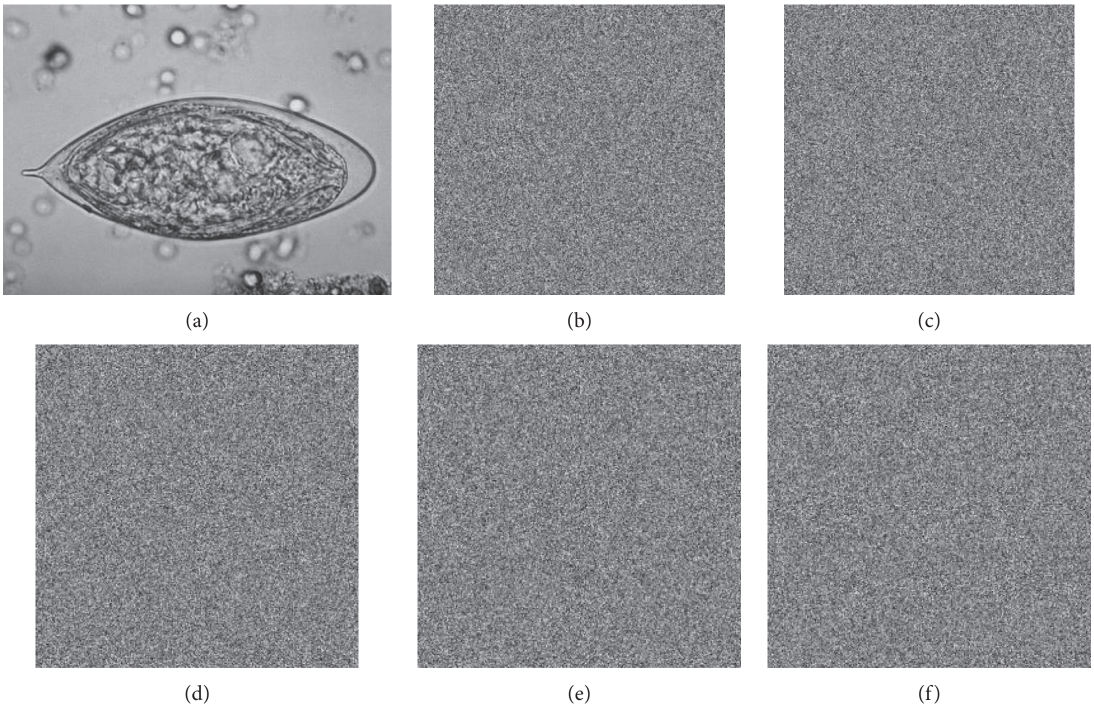


Figure 7: Plain and cipher images. (a) The plain image of «S-Hematobium egg». (b) to (f) The different cipher images obtained with different external keys.

value obtained is 0.0041. This test was performed on several other images and all the results are conclusive. This implies that the encrypted images produced from the proposed cryptosystem are different. This means that the proposed cryptosystem is very sensitive to the encryption key.

3.5.2. Key Sensitivity Test of Decryption Process. In a robust encryption scheme, an insignificant change in the key should not let to the decryption of the cipher image successfully [17]. The key sensitivity test was performed using a slightly different external key to decrypt the encrypted images. Some examples are given below. A "*ultrasound of fetus of 3 months*" image (Figure 8(a)) has been encrypted using the proposed cryptosystem where the external key is "A23C56789ABADEF7167DEAB6789367A9». The encrypted image is shown in Figure 8(b).

(a) Firstly, the encrypted image (Figure 8(b)) is decrypted with a decrypted external key «A23D56789ABADEF7167DEAB6789367A9» which is different to the encryption external key «A23C56789ABADEF7167DEAB6789367A9» by a

Table 7: Correlation coefficients between various cipher images presented in Figure 7

Images	Correlation coefficients
Figures 6(a) and 6(b)	−0.0010
Figures 6(a) and 6(c)	−6.6046e−04
Figures 6(a) and 6(d)	−2.6622e−04
Figures 6(a) and 6(e)	1.4038e−04
Figures 6(a) and 6(f)	0.0014
Figures 6(b) and 6(c)	5.5615e−04
Figures 6(b) and 6(d)	0.0031
Figures 6(b) and 6(e)	0.0027
Figures 6(b) and 6(f)	−0.0016
Figures 6(c) and 6(d)	0.0041
Figures 6(c) and 6(e)	−0.0022
Figures 6(c) and 6(f)	8.2013e−04
Figures 6(d) and 6(e)	−5.3809e−04
Figures 6(d) and 6(f)	8.4005e−04
Figures 6(e) and 6(f)	−6.3580e−04

single bit. The fourth character C in the encryption external key is changed into D in the decryption external key. The decrypted image is shown in Figure 8(c).

(b) In the second test, the encrypted image (Figure 8(b)) is decrypted by using «A23C56789AB-ADEF6167DEAB6789367A9» as the external key. The sixteenth character **7** in the encryption external key is changed into **6** in the decryption external key. The decrypted image is shown in Figure 8(d).

(c) In this case, the encrypted image (Figure 8(b)) is decrypted using the decrypted external key «A23C56789ABADEF7167DEAB6789367AA». The last character 9 in the encryption external key is changed into A in the decryption external key. Figure 8(e) shows the decrypted image which is not correlated with the original image.

(d) In Figure 8, Figure 8(b) has been decrypted using «A33C56789ABADEF7167DEAB6789367A9» as the decrypted external key. The second character 2 in the encryption external key is changed into 3 in the decryption external key.

Physically, the decrypted images are not similar to a plain image "*ultrasound of fetus of 3 months*," as we can see in Figure 8. The correlation coefficients between the plain image and the decrypted images using a slightly different key have been calculated. They are all closed to zero as we can see in Table 8. Any change on an external secret key affects the angle of the eight used rotations and changes an internal key. Consequently, the resulting substitution-diffusion key is modified. It comes out that, without an exact key, one cannot succeed in the decryption process. This confirms the effectiveness and key sensitivity of the proposed algorithm.

So, the proposed cryptosystem is very sensitive to the encryption and decryption of external keys.

3.6. Attack Analyses

3.6.1. Entropy Information Analysis. It comes out from Tables 3 and 4 that the values of entropy information obtained with our new proposed scheme on medical and nonmedical images are very close to 8. The highest value is 7.9998 and the lowest one is 7.9991 for medical images, while the lowest value is 7.9992 and the highest value is 7.9994 for nonmedical images. These values are very close to eight (ideal value) compared to the entropy of the original images. This indicates that the proposed algorithm has hidden information randomly, and information leakage in the encryption process is negligible. We evaluated the local entropy of many images using TB = 1936 as in [43], and the results are presented in Table 9. We conclude the effectiveness of the algorithm considering the high values of entropy information and the local entropy.

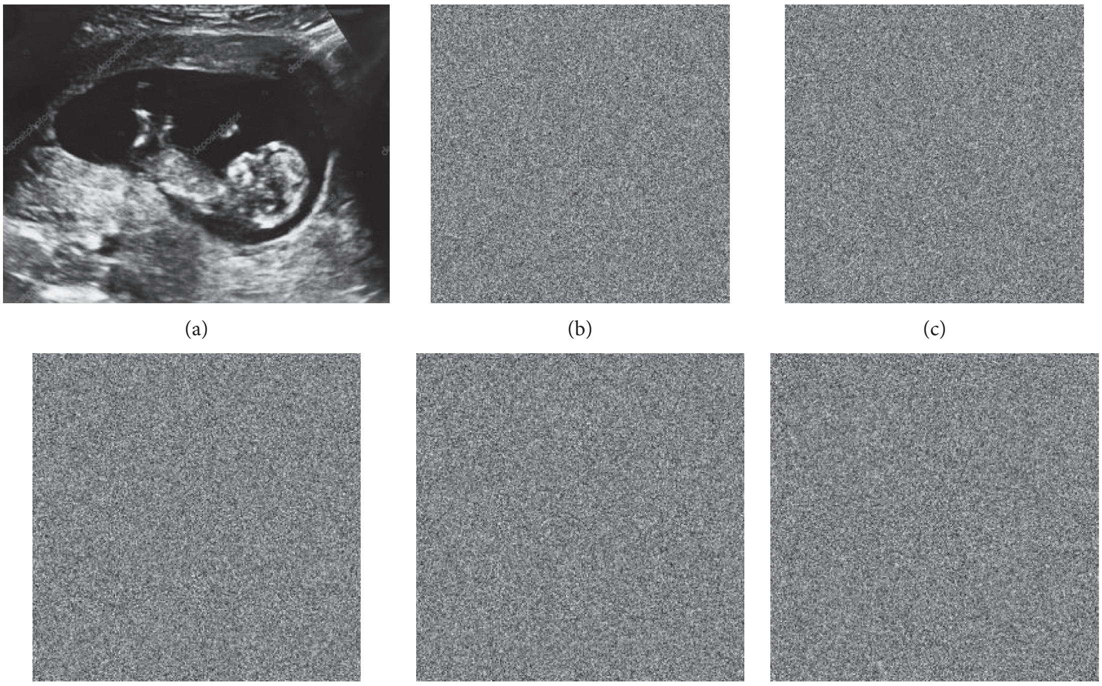
3.6.2. Differential Attacks. Tables 10 and 11 present the values of the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) for medical and nonmedical images.

In all the cases tested, the NPCR values are closed to 99.6% and the UACI is found close to 33.33%. Our new algorithm is very sensitive with respect to a small percentage of pixels' change in the plain image and the rate of influence because one-pixel change in the plain image is very high. According to Tables 10 and 11, the correlation coefficients between the cipher images are negligible. So, a minor pixel change in the plain image has an important effect on the cipher image. The proposed encryption scheme is sensitive to a minor change in the plain image.

We have evaluated MSE and PSNR on all test images; the MSE is zero in all the cases and the PSNR is infinite, as we can see in Tables 10 and 11. The original and the decrypted image are identical in all cases.

4. Discussion

This work proposes a new image encryption algorithm which does not use chaotic functions or mathematical functions which are time-consuming and make the algorithm too complex. It uses an external key of 128 bit size and an internal key. The originality of this method dwells on the combination of external and internal keys and the use of reflection or rotation mapping and the method to apply substitution and zigzag processes. An internal key comes from the decomposition of an image to be encrypted. The method to extract an internal key has been explained in Section 2. This internal key is the first level to ensure the security of the proposed system. To increase the security of this system, we combine the two keys to produce the diffusion-substitution sequences. These two aspects are the first novelty of the work. The second novelty comes from binary image processing. Each binary image is reshuffled within the


(a) (b) (c)

(d) (e) (f)

Figure 8: (a and b) The plain image from "ultrasound of fetus of 3 months" image and its corresponding encrypted image. (c)-(f) The decrypted images from the encrypted image of Figure 8(b) using slightly different decryption keys than the key used for encryption.

Table 8: Correlation coefficients between various decrypted images shown in Figure 8.

Images	Correlation coefficients
Figures 7(a) and 7(b)	0.0019
Figures 7(a) and 7(c)	−5.0690*e*−04
Figures 7(a) and 7(d)	9.6489*e*−04
Figures 7(a) and 7(e)	−5.8923*e*−04
Figures 7(a) and 7(f)	1.4796*e*−04
Figures 7(b) and 7(c)	−0.0014
Figures 7(b) and 7(d)	−0.0014
Figures 7(b) and 7(e)	8.0764*e*−04
Figures 7(b) and 7(f)	−0.0018
Figures 7(c) and 7(d)	−0.0016
Figures 7(c) and 7(e)	−1.0391*e*−04
Figures 7(c) and 7(f)	0.0030
Figures 7(d) and 7(e)	0.0038
Figures 7(d) and 7(f)	0.0024
Figures 7(e) and 7(f)	8.9324*e*−04

Table 9: Local entropies for the cipher images.

Images	Balantidium coli cyst (200 × 200)	Echopelv (601 × 711)	Girl (lena, 4.2.04) (512 × 512)	Guefack (365 × 486)
Local entropy information	7.9088	7.9081	7.9091	7.9085

block by the zigzag path. The corresponding reflections and rotations are applied to binary images obtained from the decomposition of the original image. It is also the case during the subblock substitution process. The scan zigzag process is applied not on the pixels of the subblock but on the bits of the pixel. The new pixel block is reconstructed. Consequently, the zigzag operation changes the values of the pixels and acts as a substitution process. In image encryption algorithm, the size of the external key, the size of subblock, the number of the subblock, the number of the rounds on each subblock, the variation of the key from one subblock to another one and from one round to another one, the generator used to generate sequences for the substitution-diffusion process, and the encryption scheme are the factors

Table 10: Values of the number of pixels change rate (NPCR) and unified average changing intensity (UACI) in several medical images.

Image name	Corr (A.B)	NCPR	UACI	MSE	PSNR
ANTAMOEBACOLI	−1.72*e*−03	99.6166	33.4613	0	∞
article_oeuf_taeniaC2	1.26*e*−03	99.6227	33.4476	0	∞
Balantidium coli cyst	−1.43*e*−03	99.6082	33.4647	0	∞
Balantidium coli_trophozoite	3.73*e*−04	99.6044	33.4487	0	∞
DICROCOELIUM	−5.02*e*−04	99.604	33.4573	0	∞
Entamoeba coli trophozoite	7.61*e*−04	99.6201	33.4661	0	∞
Entamoeba histolytica cyst	−1.82*e*−03	99.6105	33.5386	0	∞
Entamoeba histolytica-cyst-Gini	−2.94*e*−03	99.6159	33.5402	0	∞
Entamoeba histolytica trophozoite	6.55*e*−04	99.6124	33.4565	0	∞
Entamoeba histolytica trophozoite_redim	8.47*e*−04	99.6067	33.4592	0	∞
Entamoeba histolytica trophozoite_redim2	2.35*e*−04	99.5987	33.4662	0	∞
Oeuf_ascarisc	7.00*e*−04	99.6132	33.4818	0	∞
S-Hematobium egg	−1.13*e*−03	99.6124	33.44	0	∞
S- Mansoni egg	−1.20*e*−03	99.604	33.5229	0	∞
Tropho_entamoeba_histolytica2	1.82*e*−03	99.6132	33.4328	0	∞
Tropho_iodamoeba_butschlii	1.46*e*−04	99.5926	33.472	0	∞
Angio	5.33*e*−04	99.5983	33.4329	0	∞
DisLocElbow	−1.62*e*−03	99.6235	33.4576	0	∞
Echo1	7.11*e*−04	99.6166	33.4819	0	∞
I1_200	−3.09*e*−03	99.6059	33.5545	0	∞
Node2	9.36*e*−04	99.6075	33.4614	0	∞
Ossify	−7.55*e*−04	99.6227	33.5145	0	∞
Pelvis	−3.33*e*−03	99.6227	33.4752	0	∞
Ribs	3.25*e*−03	99.6136	33.4069	0	∞
Dirofilaria	2.97*e*−04	99.6151	33.4711	0	∞
Headirm	1.65*e*−04	99.5991	33.4599	0	∞
Abdomenirm	−2.53*e*−04	99.6212	33.478	0	∞
Pelvisirm	3.67*e*−03	99.5869	33.3766	0	∞
Gastrointestinal_parasites	−1.53*e*−03	99.6235	33.4498	0	∞
Echo fetus at 12 weeks	3.31*e*−03	99.6189	33.4343	0	∞
Ultrasound of fetus of 3 months	−2.24*e*−03	99.5979	33.5196	0	∞
Echopelv	1.79*e*−05	99.6007	33.4657	0	∞
CT-MONO2-8-abdo	−4.7542*e*−04	99.617	33.4636	0	∞
OT-MONO2-8-colon	−1.6*e*−03	99.6124	33.498	0	∞

that provide to the security, efficiency, and robustness to the method. The encryption method proposed in this work is based on these parameters; when they are not enough, one can easily cryptanalyse the cipher image during transmission. These features and the obtained results make the proposed system resistant to any kind of attack while complicating the task of cryptanalysis. Certainly, our algorithm is not very fast as the others, but we fight against cryptanalysis, and we gain on efficiency and security. Table 12 presents the comparison of the results with those obtained in [28, 33, 37, 38, 44] by presenting the correlation coefficient between vertically, horizontally, and diagonally adjacent pixels of plain and cipher image, entropy, NPCR, and UACIA of "*Lena, Airplane and Baboon*" images.

Our algorithm gives the best vertical correlation coefficient (9.09*e*−05 for "Lena" image, −0.000767 for "Airplane "image, and 0.00136 for "Baboon" image). In terms of horizontal correlation coefficient, the proposed algorithm also gives the best results (0.000257 for "Lena" image and −0.000584 for "Baboon" image). In [44], the horizontal correlation coefficient on "Airplane "image is low (0.0017 compare to ours 0.00228), but we have a high entropy value (7.994 compared to theirs 7.990). For diagonal correlation coefficient obtained in [37] is the lowest (−0.00018) on «Lena» image, but the proposed algorithm also gives the highest entropy value (7.9993 for "Lena" image, 7.9994 for "Airplane" image, and 7.9993 for "Baboon" image). In [44], the nearest value of the entropy 7.9992 has been obtained, but chaotic sequences are used and the algorithm used to obtain these sequences is too complex. In terms of NPCR and UACIA, the values are very close in all cases. According to the results, the proposed method provides better performance than the method based on CNT proposed in [33] in terms of correlation, NPCR, and UACI. Table 13 presents the comparison results in medical images. It comes from Table 13 that the proposed method provides better performance than the method based on CNT proposed in [32]. The single difference is in "OT-MONO2-8-colon" image, where the vertical correlation value (−0.0003) in [32] is higher than ours (−2,33*e*−03). It comes from this table that the proposed method has a performance similar to that achieved by other recently proposed techniques. The advantages of the proposed method are the originality, the simplicity of algorithm, the efficiency, and the lower number of the round. We use five rounds instead of sixteen as in [28].

Table 11: Values of the number of pixels' change rate (NPCR) and unified average changing intensity (UACI) in several nonmedical images.

File name	Description	Size	Corr (A.B)	NCPR	UACI	MSE	PSNR
4.1.01	Girl	256 × 256	−1.35*e*−03	99.5995	33.452	0	∞
4.1.02	Couple	256 × 256	−9.35*e*−04	99.6117	33.4913	0	∞
4.1.03	Girl	256 × 256	−1.38*e*−03	99.6178	33.5003	0	∞
4.1.04	Girl	256 × 256	5.57*e*−04	99.6136	33.4734	0	∞
4.1.05	House	256 × 256	−6.70*e*−05	99.6017	33.481	0	∞
4.1.06	Tree	256 × 256	−3.04*e*−03	99.6166	33.5354	0	∞
4.1.07	Jelly beans	256 × 256	3.16*e*−03	99.614	33.4357	0	∞
4.1.08	Jelly beans	256 × 256	2.05*e*−03	99.604	33.4272	0	∞
4.2.01	Splash	512 × 512	9.57*e*−04	99.612	33.4338	0	∞
4.2.02	Girl (tiffany)	512 × 512	2.78*e*−03	99.6105	33.3975	0	∞
4.2.03	Mandrill (a.k.a. Baboon)	512 × 512	1.94*e*−04	99.6063	33.4353	0	∞
4.2.04	Girl (lena. or lena)	512 × 512	5.32*e*−-04	99.6254	33.4604	0	∞
4.2.05	Airplane (F-16)	512 × 512	1.45*e*−03	99.604	33.4426	0	∞
4.2.06	Sailboat on lake	512 × 512	1.39*e*−04	99.609	33.4513	0	∞
4.2.07	Peppers	512 × 512	−1.55*e*−03	99.6105	33.5116	0	∞
5.1.09	Moon surface	256 × 256	2.42*e*−03	99.6166	33.434	0	∞
5.1.10	Aerial	256 × 256	2.44*e*−03	99.6262	33.4209	0	∞
5.1.11	Airplane	256 × 256	6.09*e*−04	99.6082	33.4821	0	∞
5.1.12	Clock	256 × 256	1.58*e*−03	99.5918	33.4126	0	∞
5.1.13	Resolution chart	256 × 256	−2.49*e*−03	99.6101	33.5335	0	∞
5.1.14	Chemical plant	256 × 256	3.96*e*−03	99.6346	33.343	0	∞
5.2.08	Couple	512 × 512	8.94*e*−04	99.5956	33.4183	0	∞
5.2.09	Aerial	512 × 512	4.95b03	99.6113	33.3598	0	∞
5.2.10	Stream and bridge	512 × 512	−7.71*e*−04	99.6117	33.5107	0	∞
7.1.01	Truck	512 × 512	6.92*e*−04	99.6025	33.4377	0	∞
7.1.02	Airplane	512 × 512	2.96*e*−03	99.6044	33.4076	0	∞
7.1.03	Tank	512 × 512	−6.64*e*−04	99.6159	33.4696	0	∞
7.1.04	Car and APCs	512 × 512	2.12*e*−03	99.6105	33.4304	0	∞
7.1.05	Truck and APCs	512 × 512	7.08*e*−04	99.6254	33.4041	0	∞
7.1.06	Truck and APCs	512 × 512	−4.98*e*−04	99.5941	33.5041	0	∞
7.1.07	Tank	512 × 512	7.41*e*−04	99.6178	33.4429	0	∞
7.1.08	APC	512 × 512	−1.30*e*−05	99.596	33.4592	0	∞
7.1.09	Tank	512 × 512	−6.91*e*−04	99.6185	33.4967	0	∞
7.1.10	Car and APCs	512 × 512	2.65*e*−03	99.5937	33.3973	0	∞
Boat.512	Fishing boat	512 × 512	1.10*e*−03	99.633	33.4121	0	∞
Elaine.512	Girl (elaine)	512 × 512	−3.55*e*−03	99.612	33.475	0	∞
House	House	512 × 512	2.72*e*−04	99.6174	33.4448	0	∞
Gray21.512	21-level step wedge	512 × 512	−4.65*e*−04	99.6159	33.446	0	∞
Numbers.512	256-level test pattern	512 × 512	1.07*e*−03	99.6243	33.4408	0	∞
	Azafack	398 × 512	1.97*e*−04	99.6109	33.4439	0	∞
	Guefack	365 × 486	2.72*e*−03	99.5934	33.3593	0	∞

Table 12: Comparison of results on nonmedical images.

Image	Metric	Cipher image [34]	Cipher image [38]	Cipher image [39]	Cipher image [29]	Cipher image [44]	Cipher image of our algorithm
Lena	Vert. cor	−0.0024	0.003709	0.00085	−0.0016	0.0034	9,09*e*−05
	Hor. cor	0.0076	−0.00084	0.00080	0.0031	0.0026	−0.000257
	Diag. cor	0.003	−0.00018	0.00019	0.0067	0.0019	−0.000452
	Entropy	**7.9992 < e < 7.9994**	7.99748		7.9952	7.9992	7.9993
	NPCR	93,7457	>99.6	99.6553	>96	99,6201	99.6254
	UACI	32.3899	33.4	33.3377	31,79	33,4006	33.4604

Table 12: Continued.

Image	Metric	Cipher image [34]	Cipher image [38]	Cipher image [39]	Cipher image [29]	Cipher image [44]	Cipher image of our algorithm
Airplane (512 × 512)	Vert. cor	−0.0095				−0.0036	−0.000767
	Hor. cor	0.0025				−0.0017	0.00228
	Diag. cor	0.0009				−0.0020	−9.57*e*−05
	Entropy	**7.9992 < e < 7.9994**	7.99925			7.9990	7.9994
	NPCR	93.7482	99.5252			99.6178	99.604
	UACI	32.3293	33.38			33.589	33.4426
Baboon	Vert. cor	−0.0092				−0.0019	0.00136
	Hor. cor	0.0019				−0.0014	−0.000584
	Diag. cor	0.0049				−0.0013	−0.000473
	Entropy	**7.9992 < e < 7.9994**	7.9993			7.9991	7.9993
	NPCR	93.7483	99.9935			99.6109	99.6063
	UACI	31.7602	33.69			33.4757	33.4353

Table 13: Comparison of results on medical images.

Images	References	Vert. cor	Hor. cor	Diag. cor	Entropy	NPCR	UACI
CT-MONO2-8-abdo	[32]	−0.0056	0.0132	−0.0006	7.9856 < e < 7.9992	99.6077	33.4501
	Proposed scheme	−2.57*e*−03	−3.56*e*−03	−2.56*e*−04	7.9992	99.617	33.4636
OT-MONO2-8-colon	[32]	−0.0003	0.0012	−0.0087	7.9856 < e < 7.9992	99.6082	33.462
	Proposed scheme	−2.33*e*−03	5.95*e*−04	−4.25*e*−04	7.9993	99.6124	33.498

5. Conclusion

In this work, a new image encryption algorithm has been proposed in order to secure images during transmission. The cryptosystem uses affine transformations (reflections and rotations), an external secret key of 128 bits long, and an internal secret key coming from the decomposition of the plain image, zigzag process, and substitution-diffusion processes. The particularity of this algorithm is the method to extract the internal secret keys, the use of reflection and rotation mappings on the binary images obtained from the decomposition of the plain image to be encrypted, and the zigzag process not on the gray-scale image but on the binary image and binary block, and finally the method used to combine external and internal keys to generate substitution-diffusion sequences without complex mathematical functions or complex chaotic generators. The size of an internal secret key depends on the size of the original image. The substitution process is also taken in two steps. We have evaluated the proposed algorithm on statistical analysis and key sensitivity analysis. The new proposed system is efficient and robust. The main features of the encryption scheme are its simplicity, its efficiency, and a high security order. Our method also has better confusion, diffusion, and security compared to recent methods in the literature. The combination of external and internal secret keys, the changing of the substitution-diffusion key for one subblock to another, makes the method to be robust against brute-force attacks. The newly proposed method is expected to be useful for real-time encryption and transmission of images in many domains such as telemedicine.

Data Availability

The data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Prasad and K. L. Sudha, "Chaos Image Encryption Using Pixel Shuffling," *Computer Science & Information Technology (CS & IT)*, no. 2, pp. 169–179, 2011.

[2] N. Anane, A. Mohamed, B. Hamid, I. Mohamed, and K. Messaoudi, "Rsa based encryption decryption of medical images," in *Proceedings of the 7th International Multi-Conference on Systems Signals and Devices (SSD)*, pp. 1–4, IEEE, Amman Jordan, June 2010.

[3] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer Science & Business Media, Berlin, Germany, 2012.

[4] W. C. Barker and E. B. Barker, *Recommendation for the Triple Data Encryption Algorithm (Tdea) Block Cipher*, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2012.

[5] N. Baran, "News and views: RSA algorithm in the public domain; Woz joins the inventors hall of fame; entangled photons mean faster, smaller ICs; behemoth mothballed; advanced encryption standard selected; SGI releases sdk as open source; WSDL spec released," *Dr. Dobbâs J. Software Tools*, vol. 25, no. 12, p. 18, 2000.

[6] A. Das and A. Adhikari, “An efficient multi-use multi-secret sharing scheme based on hash function,” *Applied Mathematics Letters*, vol. 23, no. 9, pp. 993–996, 2010.

[7] S. Mazloom and A. M. E. Moghadam, “Color image encryption based on coupled nonlinear chaotic map, Chaos,” *Solitons Fractals*, vol. 42, no. 3, pp. 1745–1754, 2009.

[8] N. Nithin, A. M. Bongale, and G. P. Hegde, “Image encryption based on feal algorithm,” *International Journal of Advances in Computer Science and Technology*, vol. 2, no. 3, pp. 14–20, 2013.

[9] M. Ahmad and O. Farooq, “Chaos based PN sequence generator for cryptographic applications,” in *Proceedings of the 2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, pp. 83–86, IEEE, December 2011, Aligarh, India.

[10] X. Li, C. Li, and I. K. Lee, “Chaotic image encryption using pseudo-random masks and pixel mapping,” *Signal Processing*, vol. 125, pp. 48–63, 2016.

[11] M. Ahmad, M. Z. Alam, Z. Umayya, S. Khan, and F. Ahmad, “An image encryption approach using particle swarm optimization and chaotic map,” *International Journal of Information Technology*, vol. 10, no. 3, pp. 247–255, 2018.

[12] Y. Niu, N. Ying, Z. Xuncai, and H. Feng, “Image encryption algorithm based on hyperchaotic maps and nucleotide sequences database,” *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 4079793, 9 pages, 2017.

[13] T. Adélaïde, H. Fotsin, and J. Kengne, “Image encryption algorithm based on dynamic dna coding operations and 3d chaotic systems,” *Multimedia Tools and Applications*, vol. 80, pp. 1–31, 2021.

[14] A. Telem, D. Tchiotsop, K. Thomas, F. Hilaire, and D. Wolf, “A robust chaotic and fast walsh transform encryption for gray scale biomedical image transmission,” *Signal & Image Processing: International Journal*, vol. 6, no. 3, pp. 81–102, 2015.

[15] L. Liu and S. A. Miao, “A new image encryption algorithm based on logistic chaotic map with varying parameter,” *Springer Plus*, vol. 5, p. 289, 2016.

[16] Q. Liu, P.-Y. Li, M.-C. Zhang, Y.-X. Sui, and H.-J. Yang, “A novel image encryption algorithm based on chaos maps with Markov properties,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, pp. 506–515, 2015.

[17] N. K. T. Adelaïde, M. S. Colince, K. Godpromesse, and B. F. Hilaire, “A simple and robust gray image encryption scheme using chaotic logistic map and artificial neural network,” *Advances in Multimedia*, vol. 2014, p. 13, Article ID 602921, 2014.

[18] M. Kumari and S. Gupta, “Novel image encryption scheme based on intertwining chaotic maps and RC4 stream cipher,” *3D Research*, vol. 9, p. 10, 2018.

[19] W. K. Lee, R. C. W. Phan, W. S. Yap, and B. M. Goi, “SPRING: a novel parallel chaos-based image encryption scheme,” *Nonlinear Dynamics*, vol. 92, p. 575, 2018.

[20] H. Yuan, Y. Liu, T. Lin, T. Hu, and L.-H. Gong, “A new parallel image cryptosystem based on 5D hyper-chaotic system,” *Signal Processing: Image Communication*, vol. 52, pp. 87–96, 2017.

[21] Y. Zhang, “The image encryption algorithm based on chaos and DNA computing,” *Multimedia Tools and Applications*, vol. 77, 2018.

[22] A. Daneshgar and K. Behrooz, “A self synchronised chaotic image encryption scheme,” *Signal Processing: Image Communication*, vol. 36, pp. 106–114, 2015.

[23] X. Chai, Z. Gan, Y. Chen, and X. Liu, “An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequance operations,” *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.

[24] M. Li, Y. Guo, H. Jie, and Y. Lia, “Cryptanalyse of a chaotic image encryption scheme based on permutation-diffusion structure,” *Signal Processing: Image Communication*, vol. 62, pp. 164–172, 2018.

[25] X. Wang and G. He, “Cryptanalysis on a novel image encryption method based on total shuffling scheme,” *Optics Communications*, vol. 284, no. 24, pp. 5404–5407, 2011.

[26] X. Wang, D. Luan, and X. Bao, “Cryptanalysis of an image encryption algorithm using Chebyshev generator,” *Digital Signal Processing: A Review Journal*, vol. 25, no. 1, pp. 244–247, 2014.

[27] R. Rhouma, E. Solak, and S. Belghith, “Cryptanalysis of a new substitution–diffusion based image cipher,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 7, pp. 1887–1899, 2010.

[28] N. K. Pareek, V. Patidar, and K. K. Sud, “Diffusion-substitution based gray image encryption scheme,” *Digital Signal Processing*, vol. 23, no. 3, pp. 894–901, 2013.

[29] A. Jolfaei, X. W. Wu, and V. Muthukkumarasamy, “Comments on the security of « Diffusion-substitution based gray image encryption » scheme,” *Digital Signal Processing*, vol. 32, pp. 34–36, 2014.

[30] A. Houas, Z. Mokhtari, K. E. Melkemi, and A. Boussaad, “A novel binary image encryption algorithm based on diffuse representation,” *Engineering Science and Technology, an International Journal*, vol. 19, pp. 1887–1894, 2016.

[31] Z. Mokhtari and K. Melkemi, “A new watermarking algorithm based on entropy concept,” *Acta Applicandae Mathematica*, vol. 116, no. 1, pp. 65–69, 2011.

[32] J. B. Lima, F. Madeiro, and F. J. R. Sales, “Encryption of medical images based on cosine number transform,” *Signal Processing: Image Communication*, vol. 35, pp. 1–8, 2015.

[33] J. B. Lima, E. S. D. Silva, and R. M. Campello de Souza, “Cosine transform over fields of characteristic2: fast computation and application to image encryption,” *Signal Processing: Image Communication*, vol. 54, pp. 130–139, 2017.

[34] M. H. Annaby, M. A. Rushdi, and E. A. Nehary, “Image encryption via discrete fractional fourier-type transform genereted by random matrices,” *Signal Processing: Image Communication*, vol. 49, pp. 25–46, 2016.

[35] S. Lian, J. Sun, and Z. Wang, “A block cipher based on a suitable use of chaotic standard,” *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[36] K. W. W. Wong, B. S. HungKwok, and W. ShingLaw, “A fast image encryption scheme based on chaotic standard map,” *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2006.

[37] G. Zhang and Q. Liu, “A novel image encryption method based on total shuffling scheme,” *Optics Communications*, vol. 284, pp. 2775–2780, 2011.

[38] Z. Eslami and A. Bakhshandeh, “An improvement over an image encryption method based on total shuffling,” *Optics Communications*, vol. 286, pp. 51–55, 2013.

[39] H. Zhu, C. Zhao, X. Zhang, and L. Yang, “An image encryption scheme using generalized Arnold map and affine cipher,” *Optik*, vol. 125, no. 22, pp. 6672–6677, 2014.

[40] J. Ahmad and S. O. Hwang, “A secure image encryption scheme based on chaotic maps and affine transformation,” *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13951–13976, 2016.

[41] D. Shah, T. Shah, and S. S. Jamal, “A novel efficient image encryption algorithm based on affine transformation combine

with linear fractional transformation," *Multidimensional Systems and Signal Processing*, vol. 31, pp. 1–21, 2019.

[42] X. L. Chai, Z. H. Gan, Y. Lu, M. H. Zhang, and Y. R. Chen, "A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system," *Chinese Physics B*, vol. 25, no. 10, Article ID 100503, 2016.

[43] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm," *Optics and Lasers in Engineering*, vol. 128, Article ID 105995, 2020.

[44] J. S. Fouda, J. Y. Effa, S. Sabat, and M. Ali, "A fast chaotic block cipher for image encryption," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 3, pp. 578–588, 2014.

[45] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption. cyber journals: multidisciplinary journals in science and technology," *Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[46] Cornell University, "Vision and Image Analysis Group," 2019, http://www.via.cornell.edu/databases.

[47] barre.com, "Medical Image Samples," 2014, http://www.barre.nom.fr/medical/samples.

[48] A. Weber, "The USC-SIPI Image database," 1977, http://sipi.usc.edu/database.