IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 20, 2019, accepted March 14, 2019, date of publication March 20, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906292

Novel Medical Image Encryption Scheme Based on Chaos and DNA Encoding

AKRAM BELAZI[1], MUHAMMAD TALHA[2], (Member, IEEE), SOFIANE KHARBECH[3], (Member, IEEE), AND WEI XIANG[4], (Senior Member, IEEE)

[1]RISC Laboratory, National Engineering School of Tunis, Tunis El Manar University, Tunis 1002, Tunisia
[2]Deanship of Scientific Research, King Saud University, Riyadh 11543, Saudi Arabia
[3]Laboratory Sys'Com-ENIT (LR-99-ES21), Tunis El Manar University, Tunis 1002, Tunisia
[4]College of Science and Engineering, James Cook University, Cairns, QLD 4878, Australia

Corresponding author: Wei Xiang (wei.xiang@jcu.edu.au)

This work was supported by the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia.

ABSTRACT In this paper, we propose a new chaos-based encryption scheme for medical images. It is based on a combination of chaos and DNA computing under the scenario of two encryption rounds, preceded by a key generation layer, and follows the permutation-substitution-diffusion structure. The SHA-256 hash function alongside the initial secret keys is employed to produce the secret keys of the chaotic systems. Each round of the proposed algorithm involves six steps, i.e., block-based permutation, pixel-based substitution, DNA encoding, bit-level substitution (i.e., DNA complementing), DNA decoding, and bit-level diffusion. A thorough search of the relevant literature yielded only this time the pixel-based substitution and the bit-level substitution are used in cascade for image encryption. The key-streams in the bit-level substitution are based on the logistic-Chebyshev map, while the sine-Chebyshev map allows producing the key-streams in the bit-level diffusion. The final encrypted image is obtained by repeating once the previous steps using new secret keys. Security analyses and computer simulations both confirm that the proposed scheme is robust enough against all kinds of attacks. Its low complexity indicates its high potential for real-time and secure image applications.

INDEX TERMS Image encryption, medical images, permutation and diffusion, S-box, chaos, DNA encoding, SHA-256 hash function.

I. INTRODUCTION

The rapid progress in communications networks has led to ever increasing amounts traffic of multimedia data (i.e., images, audio, and video) over unsecured communications channels. Generally, user data contain both confidential and private information, so their security has become indispensable for protecting users from a variety of malicious attacks, avoiding loss of information, and guaranteeing integrity. As of now, many technologies are applicable for ensuring a high level of security of medical images, such as steganography [1], [2], watermarking [3], [4], and encryption [5]–[8]. In fact, to transform an original image into an unrecognizable one, encryption techniques can be employed. Encryption of medical images (e.g., mammograms, MRI, Chest X-rays, CT scans, etc.) is one of the most convenient strategies to protect the security of patients' personal information over public networks against malicious attacks. Since medical images are the private data of patients, ensuring their secure storage and transmission has become an important issue for medical applications in real-world problems [5]–[7], [9], [10]. In the last decades, chaotic cryptosystems have been much studied on account of their random behavior, ergodicity, and sensitivity to secret keys, a result of the initial conditions and control parameters [11], [12]. Therefore, chaotic systems fulfill the classic Shannon requirements regarding confusion and diffusion [13], and are well suited for cryptography problems, such as image encryption. In addition, the structure of a chaos-based encryption scheme has a low level of complexity, while ensuring a high level of security. Therefore, many chaotic image encryption schemes have been proposed [9], [14]–[20].

In [15], Zhou et *al.* proposed a simple and effective chaos-based encryption scheme using a combination of logistic

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

and tent maps, namely the Logistic-Tent System (LTS). Firstly, a random pixel is inserted in the beginning of each row in the original image under consideration. So, each row is divided into a 1D matrix, and a substitution process, based on the LTS system, is applied to each row. Thereafter, all 1D matrices are combined into a 2D matrix according to their row positions in the original image. Then, the obtained 2D matrix is rotated 90 degrees counter-clockwise. The encrypted image is finally obtained by repeating these operations four times.

A new image encryption scheme termed the Tent-Logistic map-based data encryption algorithm (TL-DEA) is proposed in [16]. The scheme achieves two rounds of an SP-network on each pixel. An original image is divided into data blocks of a fixed length. Then, substitution and permutation processes are performed on each block, using the Tent-Logistic map. All encrypted blocks are combined to obtain the encrypted image.

In [18], a new chaos-based encryption scheme is proposed. It is composed of two layers, i.e., a diffusion layer followed by a bit-permutation layer. The diffusion process, repeated r_d times, is based on a binary matrix of size 32 × 32. The permutation is done using a new modified 2-D cat map, running at the data bit level, and it retrieves its dynamic key (i.e., initial condition and control parameter) from a chaotic generator. This is iterated r_p times. The entire process is repeated r times until it reaches the required security level.

Recently, a new medical image encryption scheme based on the edge maps of a source image is designed in [9]. It is dubbed EMMIE and consists of three phases, namely bit-plane decomposition, generation of a random sequence, and permutation. A reversible decomposition method is first applied to the input medical image in order to produce some bit-planes. Then, the edge maps (i.e., binary matrices with the same size as the original bit-planes) obtained from the source image are XORed with the original bit-planes. Finally, the bit positions of all the obtained bit-planes are shuffled and then collected together with a pixel diffusion operation, which generates the encrypted image.

Wu *et al.* [19] combine the chaotic tent maps (CTM) with the rectangular transform (RT) for an image encryption scheme design. The scheme is composed of t pixel-permutation rounds, followed by a pixel-diffusion layer. The pixel permutations are performed using the enhanced two-dimensional rectangular transform, while the pixel-diffusion layer is controlled by chaotic tent maps.

Furthermore, DNA (deoxyribonucleic acid) computing technology has recently been improved and used in several fields, especially in cryptography. Thanks to its good features, such as massive parallelism, huge storage, and ultra-low power consumption [21], [22], it is suitable in the context of cryptography applications. Therefore, a variety of encryption schemes merging chaos systems with DNA encoding have been proposed in recent years [23]–[30]. For example, Wang *et al.* proposed a new image encryption scheme based on DNA sequence operations and chaotic systems [27]. The scheme follows the diffusion–permutation architecture. The diffusion works on the pixels, and is controlled by pseudorandom sequences derived from a spatiotemporal chaos system, namely, the CML (coupled map lattice). After being DNA encoded, the confused image is employed to update the initial conditions of the CML, making the cryptosystem robust against known-plaintext and chosen-plaintext attacks. By using the new initial conditions, DNA-level permutation operating both on the rows and columns is performed. The permuted DNA matrix is then confused once again and DNA decoded to obtain the encrypted image.

In addition, a secure and efficient image encryption scheme based on self-adaptive permutation–diffusion and DNA random encoding is proposed in [23]. This scheme is composed of n rounds, each of them consisting of four phases, namely DNA random encoding, self-adaptive permutation, self-adaptive diffusion, and DNA random decoding. The permutation and diffusion procedures both work on the pixels, governed by the hyper-chaotic Lorenz system, and supported by a quantization process. The cryptosystem has good immunity against plaintext attacks, as the quantization processes are disturbed by the intrinsic features of the original image.

Unlike standard images, medical images have a distinctive feature, i.e., they contain more than 70% of 0's bits [31]. Therefore, in the original image, the higher bit-planes are quite similar to the lower ones, which makes vulnerable the image cryptosystems with high concentration to higher bit-planes as the lower bit-planes contain significant information [32]. From the basis, the pixel modification in [33] and [34] becomes low efficiency in the encryption process. Broadly, the high ratio of 0's bits, downgrades the encryption impact of the permutation and substitution operations. Accordingly, there is an urgency of an appropriate and efficient image cryptosystem to accommodate the challenge of 0's bits in medical images. Besides, it should respond to the low complexity and high efficiency required by telemedicine applications.

In this context, a new medical image cryptosystem is proposed through this paper. The main architecture of this cipher consists of block-based permutation, pixel-based substitution, DNA encoding, bit-level substitution (i.e., DNA complementing), DNA decoding, and bit-level diffusion. It is clear that the proposed architecture fulfills the requirements of a strong cipher; confusion and diffusion. The major contributions of this paper are four-fold as summarized below. Firstly, to the best of our knowledge, this is the first time that the pixel-based substitution and the bit-level substitution are used in cascade for image ciphering. Secondly, the secret keys of the proposed cipher are generated by the 256-bit long hash value that depends on the original image; such dependency spreads minor changes applied in the original image or the initial keys to the entire encrypted image and then ensured a high resistance against known/chosen plaintext attacks. Thirdly, the major part of the cipher consists of permutation and substitution process, which are of low complexity time. Fourthly, instead of higher-dimensional chaotic systems, the keystreams are generated by two enhanced 1-D chaotic maps [35], where excellent chaotic performances are obtained, and the famed constraints of 1D chaotic systems are surpassed. Hence, the time complexity is further reduced.

The proposed algorithm consists of two rounds, preceded by a key generation layer, and follows the permutation-substitution-diffusion structure. In the first round, the considered secret keys (i.e., initial conditions and control parameters) of the chaotic systems are generated and controlled by initial secret keys and the 256-bit long hash value of the original image. The SHA-256 hash value of the pixels' sum of the original image alongside the new secret keys are employed to produce the secret keys of the second round. Therefore, the resultant key-streams are correlated with both the initial secret keys and the input image, which are highly sensitive to the flip a single bit in either the input image or in the secret keys. The steps of the proposed algorithm are given as follows. A block-based permutation procedure is first applied. Thereafter, the permuted blocks are combined, decomposed into four parts, and replaced by S-boxes, as in [36]. Then, these blocks are combined, converted into binary form, DNA encoded, and complemented by a binary matrix produced using the logistic-Chebyshev map. Subsequently, the complemented matrix is DNA decoded and then encrypted by a bit-level diffusion layer. The diffusion layer operates using a chaotic matrix derived from the sine-Chebyshev map. By means of second round's secret keys, the aforementioned operations are performed to obtain the final encrypted image. The proposed scheme is shown to be sufficiently fast, while maintaining a high level of security. Simulation results are provided to demonstrate the effectiveness of the proposed encryption scheme compared to some existing schemes.

The remainder of this paper is organized as follows. The preliminaries, including DNA encoding and chaotic systems, are presented in Section II. Section III describes the proposed scheme in detail, followed by an analysis of the proposed scheme in Section IV. Finally, concluding remarks are drawn in Section V.

II. PRELIMINARIES

A. DNA ENCODING

DNA is a molecule that contains the genetic information used in the growth, development, functioning, and reproduction of any living organism and diverse viruses. A DNA sequence comprises four nucleic acid bases, i.e., A (Adenine), G (Guanine), C (Cytosine), and T (Thymine). These DNA bases follow the Watson–Crick [37] principle. That is, A and T are complementary, and C and G are complementary. Usually, the four DNA bases A, C, G, and T are encoded by two bits, i.e., 00, 01, 10, and 11. In binary encoding, 0 and 1 are complementary, so 00 and 11 are complementary, as are 01 and 10. By using four bases A, C, G and T to encode 00, 01, 10 and 11, there are 24 encoding rules, out of which there are only eight rules (see Table 1) satisfying the complementary relations among the bases. DNA decoding rules are the reverse of its encoding counterparts. For example, if the grayscale value of a pixel is 100, its corresponding binary value is "10011101", which can be encoded as the DNA sequence "GCAC" using DNA encoding Rule 7.

TABLE 1. DNA encoding rules.

	Rule							
Base	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
C	01	10	00	11	00	11	01	10
G	10	01	11	00	11	00	10	01

B. LOGISTIC-CHEBYSHEV MAP

The logistic-Chebyshev map (denoted by LCv) is an enhanced one-dimensional (1D) chaotic system that combines two well-known seed maps, namely the logistic map and the Chebyshev map [35]. It is defined as follows

$$x_{n+1} = \left(\alpha x_n\left(1 - x_n\right) + \frac{(4-\alpha)}{4} \cos\left(b.\arccos(x_n)\right)\right) \bmod 1 \tag{1}$$

where $x_n \in (0, 1)$ is the initial value and $\alpha \in (0, 4]$ is a control parameter. The degree of the Chebyshev map is $b \in \mathbb{N}$.

C. SINE-CHEBYSHEV MAP

The sine-Chebyshev map (denoted by SCv) is an improved one-dimensional (1D) chaotic system that combines two well-known seed maps, namely the sine map and the Chebyshev map [35]. It is defined as follows

$$x_{n+1} = \left(\alpha \sin\left(\pi x_n\right) + \frac{(4-\alpha)}{4} \cos\left(b.\arccos(x_n)\right)\right) \bmod 1 \tag{2}$$

where $x_n \in (0, 1)$ is the initial value, $\alpha \in (0, 4]$ is a control parameter, and $b \in \mathbb{N}$ is the degree of the Chebyshev map.

III. PROPOSED MEDICAL IMAGE ENCRYPTION APPROACH

A. PERMUTATION FUNCTION AND ITS INVERSE

In modular arithmetic, when p is a prime number then it is divisible by any non-zero number. Thus, for each $x \in \{1, 2, \ldots, n-1\}$ there is one and only one number $y \in \{1, 2, \ldots, n-1\}$ such that

$$(x \times y) \bmod p = 1. \tag{3}$$

The result given by (3) is derived from the Fermat's little theorem. It follows from (3) that we define the following a permutation function

$$\mathbf{b} = (p \times \mathbf{a}) \bmod l \tag{4}$$

where $\mathbf{a} = \{1, \ldots, l\}$, l is an integer and p is a prime number considered to be the key of (4). This function allows the generation of a random sequence $\mathbf{b}^{1\times l}$ from $\{1, \ldots, l\}$. The sequence $\mathbf{a}$ can be recovered as $\mathbf{a} = (p \times \mathbf{b}) \bmod l$.

Let p be a prime number and $\mathbf{P}$ be an $M \times N$ image partitioned into l blocks of size $m \times n$, $\mathbf{D}^{m\times n}$. The block with index i is denoted by $\mathbf{D}_i^{m\times n}$, where $i = 1, 2, \ldots, l$. Now, we use (4) to calculate the new indices $\mathbf{b}$ of these blocks. Therefore, the permuted blocks $\mathbf{F}^{m\times n}$ are obtained as follows

$$\mathbf{F}_i = \mathbf{D}_{\mathbf{b}(i)}. \tag{5}$$

By merging $\mathbf{F}^{m\times n}$ into an $M \times N$ matrix, we obtain the permuted image $\mathbf{C}$. Fig. 1 shows the original image of XA Coronary and its permuted versions generated by applying (4) and (5) with different block sizes.

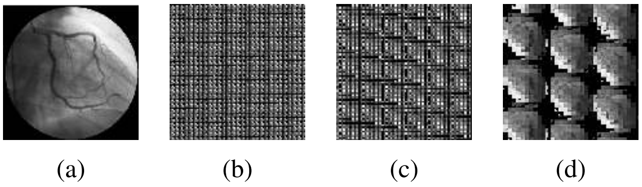

(a) (b) (c) (d)

FIGURE 1. Original image of XA Coronary (a) and its permuted versions (b), (c), and (d) where the block sizes are equal to 4 × 4, 8 × 8, and 16 × 16, respectively.

B. GENERATION OF THE INITIAL VALUES AND THE CONTROL PARAMETERS OF THE CHAOTIC MAPS

SHA-256 is a commonly used cryptographic hash function with 256-bit hash value [38]. For two images with only one-bit difference, to produce entirely different secret keys of the chaotic systems (1) and (2), the SHA-256 hash values of the original image $\boldsymbol{\kappa}_1$ and that of its sum of pixels $\boldsymbol{\kappa}_2$ are employed. In the proposed algorithm, the secret keys (i.e., initial conditions and control parameters) (x_0, α_0) and (x_1, α_1) of (1) and (2), respectively, are updated by means of the SHA-256 hash function. We first compute the SHA-256 hash value $\boldsymbol{\kappa}_1$ and then divide it into 8-bit blocks as follows

$$\boldsymbol{\kappa}_1 = \boldsymbol{\kappa}_1(1), \boldsymbol{\kappa}_1(2), \ldots, \boldsymbol{\kappa}_1(32). \tag{6}$$

The secret keys are updated as follows

$$x_0' = \frac{1}{3}\left(x_0 + 2 \times \frac{1}{256} \times \texttt{bin2dec}\,(\boldsymbol{\kappa}_1(9) \oplus \ldots \oplus \boldsymbol{\kappa}_1(16))\right) \tag{7}$$

$$\alpha_0' = \frac{1}{3}\left(\alpha_0 + 2 \times \frac{1}{64} \times \texttt{bin2dec}\,(\boldsymbol{\kappa}_1(1) \oplus \ldots \oplus \boldsymbol{\kappa}_1(8))\right) \tag{8}$$

$$x_1' = \frac{1}{3}\left(x_1 + 2 \times \frac{1}{256} \times \texttt{bin2dec}\,(\boldsymbol{\kappa}_1(1) \oplus \ldots \oplus \boldsymbol{\kappa}_1(32))\right) \tag{9}$$

$$\alpha_1' = \frac{1}{3}\left(\alpha_1 + 2 \times \frac{1}{64}\left(\sum_{i=1}^{32} \boldsymbol{\kappa}_1(i) \bmod 256\right)\right) \tag{10}$$

where $\oplus$ denotes the bit-XOR operation, and `bin2dec()` is a function that converts a binary string to the corresponding decimal number.

Secondly, using the updated parameters alongside the SHA-256 hash value $\boldsymbol{\kappa}_2$, we calculate the new parameters x_0'', α_0'', x_1'', and α_1'' using (6)-(10).

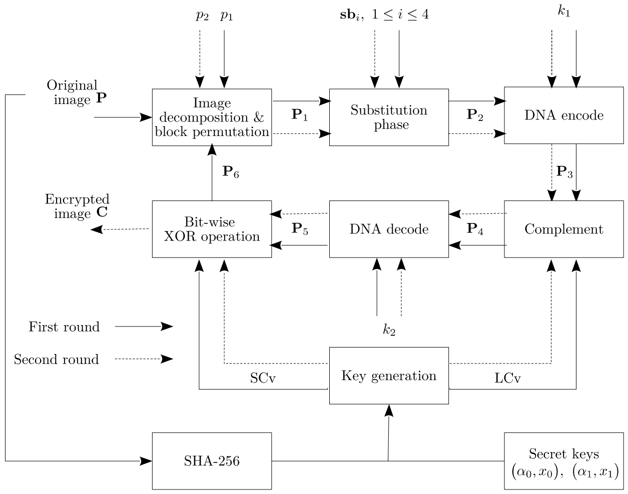


FIGURE 2. Flowchart of the proposed encryption approach.

C. ENCRYPTION PROCESS

The detailed steps of the proposed medical encryption approach, illustrated in Fig. 2, are given below.

- **Step 1:** Input 8-bit gray image $\mathbf{P}^{M\times N}$, where $M \times N$ are the dimensions of the rows and columns of the image. Also, let (x_0, α_0) and (x_1, α_1) be the secret keys (i.e., initial conditions and control parameters) of (1) and (2), respectively. Moreover, select two prime numbers p_1 and p_2;
- **Step 2:** Update the secret keys (x_0', α_0'), (x_1', α_1'), (x_0'', α_0'') and (x_1'', α_1'') through (6)-(10);
- **Step 3:** Decompose image $\mathbf{P}$ into $m \times n$ blocks, i.e., $\mathbf{H}^{m\times n}$. Then, the number of $m \times n$ blocks in $\mathbf{P}$ is $v = \frac{M\times N}{m\times n}$. $\mathbf{H}_i$ denotes the ith block, where $i = 1, 2, \ldots, v$. Next, we permute these blocks using (4)-(5) with key p_1. By merging the permuted blocks into an $M \times N$ matrix, we obtain the permuted image $\mathbf{P}_1$;
- **Step 4:** Divide the permuted image $\mathbf{P}_1$ into four blocks $\{\mathbf{B}_i, i = 1, \ldots, 4\}$. Assume that the total number of pixels in each block is k. Denote by $\left\{p_j, j = 1, \ldots, k\right\}$ the j^{th} pixel in block $\mathbf{B}_i$. Then, transform the blocks $\{\mathbf{B}_i, i = 1, \ldots, 4\}$ using $(\mathbf{sb}_i)_{i=1,2,3,4}$, which are the proposed S-boxes in [36] (denoted by S-box-3, S-box-4, S-box-5 and S-box-6, respectively). This process is as follows:
  Let $s_j = \mathbf{sb}_i(p_j + 1)$, where $i = 1, \ldots, 4$ and $j = 1, \ldots, k$. s_j are the new pixels and p_j are replaced by the S-box $\mathbf{sb}_i$. In this way, we obtain a new matrix $\mathbf{P}_2$;
- **Step 5:** Transform $\mathbf{P}_2$ into its binary form $\mathbf{P}_b$, a matrix of dimension $M \times 8N$. Thereafter, we apply the DNA encoding rule k_1 $\big($this integer is randomly selected from $\{1, \ldots, 8\}\big)$ to $\mathbf{P}_b$. Then, a $M \times 4N$ DNA matrix $\mathbf{P}_3$ is obtained;
- **Step 6:** Iterate (1) for l ($l \geq 500$) times to hide the transient effect using the updated initial condition x_0' and control parameter α_0'. Continue to iterate (1) for $M \times 4N$ to obtain the chaotic matrix $\mathbf{X}^{M\times 4N}$. Then, the map of $\mathbf{X}$

from [0, 1] to {0, 1} is defined by (11)

$$\mathbf{X}(i,j)=\begin{cases}1 & \text{if } \mathbf{X}(i,j)>0.5,\\ 0 & \text{if } \mathbf{X}(i,j)\leq 0.5.\end{cases} \tag{11}$$

It was mentioned above that in DNA code, A and T are always complementary, as are C and G. Thus, to calculate the complementary matrix of $\mathbf{P}_3$ according to the values of $\mathbf{X}$, we proceed as follows

$$\mathbf{P}_4(i,j)=\begin{cases}\mathbf{P}_3(i,j) & \text{if } \mathbf{X}(i,j)=0,\\ \overline{\mathbf{P}_3(i,j)} & \text{if } \mathbf{X}(i,j)=1.\end{cases} \tag{12}$$

- **Step 7:** Decode $\mathbf{P}_4$ using the DNA decoding rule k_2 (an integer randomly selected from $\{1,\ldots,8\}$) to obtain the decoded $M\times 8N$ matrix $\mathbf{P}_{de}$. Next, carry out decimal conversions on $\mathbf{P}_{de}$ to have an $M\times N$ matrix $\mathbf{P}_5$;
- **Step 8:** Iterate (2) for l ($l\geq 500$) times to avoid the transient effect with the initial condition x_1' and control parameter α_1'. Next, generate a chaotic $M\times N$ matrix $\mathbf{Y}$ by continuing to iterate (2) for $M\times N$ times. Then, map $\mathbf{Y}$ from [0, 1] to $\{0,1,2,\ldots,255\}$ according to the following

  $$\mathbf{U}=\left(\mathbf{Y}\times 10^{15}\right) \bmod 256.$$

  A scrambled matrix is generated as follows

  $$\mathbf{P}_6=\mathbf{P}_5\oplus\mathbf{U}. \tag{13}$$

- **Step 9:** After replacing p_1 with p_2, we repeat once Steps (3)–(8) using (x_0'',α_0'') and (x_1'',α_1'') as secret keys for (1) and (2), respectively. Thus the encrypted image $\mathbf{C}$ is obtained.

D. DECRYPTION PROCESS

The decryption procedure is performed in the reverse order of its encryption counterpart, described previously. The SHA-256 hash values κ_1 and κ_2, the prime numbers p_1 and p_2, the integers k_1 and k_2, and the S-boxes $(\mathbf{sb}_i)_{i=1,2,3,4}$ should be transmitted to the decryption side. The receiver can now produce the required parameters of (1) and (2), in an effort to recover the original image properly. The decryption process, illustrated in Fig. 3, is described in details by the following steps:

- **Step 1:** Perform bit-wise XOR operation between the encrypted image $\mathbf{C}$ and $\mathbf{V}$, which is the matrix yielded in Step 8 of the encryption process using (x_1'',α_1'') as the secret key of (2). This yields matrix $\mathbf{C}_1$;
- **Step 2:** Convert $\mathbf{C}_1$ to binary form $\mathbf{C}_b$, which is of dimension $M\times 8N$. Next, encode $\mathbf{C}_b$ using the DNA encoding rule k_2 and save the encoded $M\times 4N$ matrix $\mathbf{C}_2$;
- **Step 3:** Calculate the complementary matrix of $\mathbf{C}_2$ according to matrix $\mathbf{Z}$ as follows

  $$\mathbf{C}_3(i,j)=\begin{cases}\mathbf{C}_2(i,j) & \text{if } \mathbf{Z}(i,j)=0,\\ \overline{\mathbf{C}_2(i,j)} & \text{if } \mathbf{Z}(i,j)=1,\end{cases} \tag{14}$$

  where $\mathbf{Z}$ is generated in Step 6 of the encryption process using (x_0'',α_0'') as the secret key of (1);

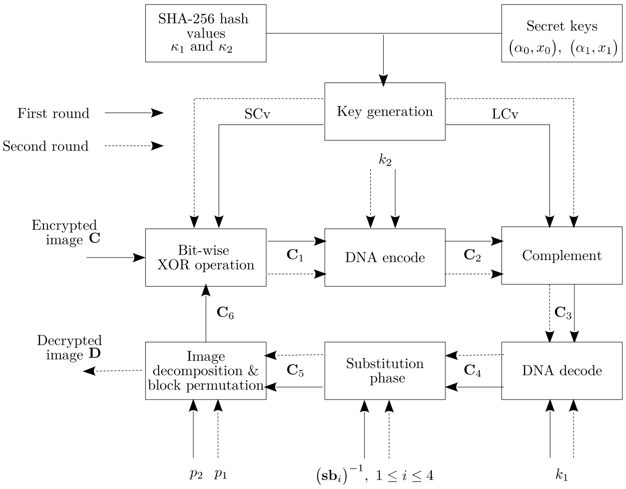

SHA-256 hash values κ_1 and κ_2
Secret keys (α_0,x_0), (α_1,x_1)
First round
Second round
Key generation
SCv
LCv
k_2
Encrypted image $\mathbf{C}$
Bit-wise XOR operation
$\mathbf{C}_1$
DNA encode
$\mathbf{C}_2$
Complement
$\mathbf{C}_6$
$\mathbf{C}_3$
Decrypted image $\mathbf{D}$
Image decomposition & block permutation
$\mathbf{C}_5$
Substitution phase
$\mathbf{C}_4$
DNA decode
p_2 p_1
$(\mathbf{sb}_i)^{-1}$, $1\leq i\leq 4$
k_1

FIGURE 3. Flowchart of the proposed decryption approach.

- **Step 4:** Apply the DNA decoding rule k_1 to $\mathbf{C}_3$. Then, the decoded matrix $\mathbf{C}_4$, of dimension $M\times 8N$, is yielded;
- **Step 5:** Convert $\mathbf{C}_4$ to decimal form $\mathbf{C}_d^{M\times N}$. Next, we transform $\mathbf{C}_d$ in Step 4 of the encryption process but with the inverse S-boxes of $(\mathbf{sb}_i)_{i=1,2,3,4}$, denoted by $(\mathbf{sb}_i)_{i=1,2,3,4}^{-1}$. The output of this step is a matrix denoted by $\mathbf{C}_5$;
- **Step 6:** Apply Step 3 of the encryption process to $\mathbf{C}_5$, where p_2 is the key of (4). This yields the permuted matrix $\mathbf{C}_6$;
- **Step 7:** After replacing p_2 with p_1, we carry out Steps (1)–(6) once again using (x_0',α_0') and (x_1',α_1') as secret keys for (1) and (2), respectively. This yields the decrypted image $\mathbf{D}$.

IV. PERFORMANCE ANALYSIS

In this section, we evaluate the security and performance of the proposed encryption approach against the results of several tests in [14], [16], and [20]. Many standard and medical 8-bit grayscale images [39], [40] of size 512 × 512 are employed as the original images. The initial conditions (x_0,x_1) are fixed at (0.3, 0.6), while the control parameters (α_0,α_1) are chosen to be (3.9, 3.9). Integers m and n are both set to 4, and the prime numbers p_1 and p_2 are fixed at 29 and 41, respectively. The DNA encoding and decoding keys k_1 and k_2 are chosen to be 7 and 1, respectively. Considering that they are recently proposed and contain significant contributions overcome most of the previous algorithms, some state-of-the-art algorithms from [14], [16], [20] are compared to the proposed algorithm.

A. KEY SPACE ANALYSIS

The key space is defined as the entire collection of the keys that could be used in the encryption process. The key space of a good image encryption scheme should be sufficiently

TABLE 2. Percentage of '1's in different original images vs. their encrypted counterparts in the first and second rounds.

Image	Stage	8th bit	7th bit	6th bit	5th bit	4th bit	3rd bit	2nd bit	1st bit
CR Chest	Original (%)	50.7908	35.1303	51.2142	50.3551	46.9212	45.7890	52.6299	48.4497
	1st round (%)	49.9928	50.0042	50.1945	50.0324	50.1026	50.0893	50.0324	49.8772
	2nd round (%)	49.9374	50.0412	50.0458	50.0320	50.1202	49.8005	49.9447	49.9805
CT Abdomen	Original (%)	11.0600	25.2174	32.4394	28.9879	29.5914	29.9191	30.1228	30.1891
	1st round (%)	50.0374	50.2159	49.9302	49.9928	49.9241	50.0546	49.9699	50.0561
	2nd round (%)	50.1209	50.0938	49.9790	49.9187	49.9107	49.9947	49.9050	50.0340
CT Cardiac	Original (%)	37.7037	23.3601	33.5407	31.9805	31.8172	32.2231	32.6347	34.3609
	1st round (%)	50.0889	49.8535	49.9046	50.0214	50.2144	49.9702	50.0763	49.8928
	2nd round (%)	49.8058	50.0370	50.0748	49.9893	49.9592	49.9893	49.9901	50.0187
CT Dental	Original (%)	29.2637	48.5558	25.1713	32.1152	34.0210	34.0282	33.8856	34.1728
	1st round (%)	50.1980	50.0751	49.9706	50.1301	49.8367	50.1369	50.0591	49.9199
	2nd round (%)	50.0732	49.9874	49.9844	49.9184	50.0248	49.9767	50.0103	50.0450
CT Head	Original (%)	29.3407	48.4165	27.9808	31.1665	50.8121	54.6913	29.1225	47.0764
	1st round (%)	49.9023	50.0031	49.9783	50.1186	50.1503	49.9645	50.0557	50.1865
	2nd round (%)	50.0881	50.0431	49.9226	50.0328	50.1465	49.9519	50.0160	50.0973
CT Paranasal sinus	Original (%)	0	0	0	13.2351	21.8407	22.7547	21.2223	15.2760
	1st round (%)	49.9516	49.9222	50.0809	49.8459	50.0172	50.0511	49.8699	49.9737
	2nd round (%)	50.0542	49.9107	50.1392	50.0156	49.9023	50.0755	50.1438	50.0011
MR Shoulder	Original (%)	10.3550	08.4568	21.7403	29.6871	33.5720	41.2113	39.7274	39.9857
	1st round (%)	50.0496	50.2533	49.9794	50.0195	49.9554	50.1106	50.0690	50.0984
	2nd round (%)	49.9538	50.0385	50.0134	50.0530	50.0271	49.9226	49.9584	49.8775
XA Coronary	Original (%)	22.4335	44.3417	40.8356	37.3009	37.1246	36.7046	37.4489	38.4102
	1st round (%)	49.9664	50.0427	50.0679	50.0393	50.0404	49.9271	49.9756	49.9905
	2nd round (%)	50.1198	50.1698	49.9897	50.0965	49.9817	49.9008	49.9584	50.0381

large to make brute-force attacks intractable. In the proposed cryptosystem, the key space includes: the initial conditions (x_0, x_1), the control parameters (α_0, α_1), the prime numbers p_1 and p_2, the integers m and n, the S-boxes $(\mathbf{sb}_i)_{i=1,2,3,4}$, the DNA encoding key k_1, and the DNA decoding key k_2. Here, $x_0, x_1 \in (0, 1)$; $\alpha_0, \alpha_1 \in (0, 4]$; p_1 and p_2 are prime numbers with an infinite number of values that can be used in a block permutation phase. The integers m and n are chosen to satisfy $(M \times N) \bmod (m \times n) = 0$, while k_1 and k_2 are two integers randomly selected from $\{1, \ldots, 8\}$, so that there are 64 possible combinations of (k_1, k_2). Moreover, each S-box $(\mathbf{sb}_i)_{i=1,2,3,4}$ used in Step 4 of the encryption process is generated by means of a logistic map [36] with the initial condition y_0^i and the control parameter β_0^i, where $i = 1, 2, 3, 4$.

It is assumed that the computational precision of the 64-bit double-precision numbers is 2^{-49}. Therefore, the total number of values of x_0 is larger than 2^{49}, as are the numbers of $x_1, y_0^1, y_0^2, y_0^3, y_0^4, \alpha_0, \alpha_1, \beta_0^1, \beta_0^2, \beta_0^3$, and β_0^4. Moreover, the security of SHA-256 with complexity of the best attack is 2^{128}. Consequently, the approach possesses more than 2^{716} secret keys, indicating a key length greater than $\log_2(2^{716}) = 716$ bits. Hence, the key space is sufficiently large to resist all presently known brute-force attacks.

B. STATISTICAL ANALYSIS

1) UNIFORMITY OF THE BIT DISTRIBUTION WITHIN EACH BIT-PLANE

Zhang *et al.* [34] demonstrated that in an original image, the higher bit planes are very correlated, particularly the 7th and 8th bit planes. This fact can be exploiting by hackers to recover a significant percentage of bits in a higher bit plane, when knowing its neighboring bit plane. Therefore, bit distribution within each bit plane should be uniform, resulting in a bit-uniformity rate of 50%. As can seen from Table 2, the bit distribution within the encrypted images' bit planes (in the first and second rounds) are more uniform than those of the original images.

2) CORRELATION OF ADJACENT PIXELS

In this section, the correlation between adjacent pixels in the original image and their respective encrypted images is studied. In original images, the value of a pixel is very close to the values of its horizontally, vertically, and diagonally adjacent pixels. As a result, the adjacent pixels in an original image are highly correlated. A cryptanalyst can exploit this weakness to break encryption. Therefore, in a ciphered image, adjacent pixels should be uncorrelated. The correlation between pairs of adjacent pixels can be calculated as follows

$$r_{xy} = \frac{\sum_{i=1}^{N} (x_i - \mathbb{E}\{x\})\,(y_i - \mathbb{E}\{y\})}{\sqrt{\sum_{i=1}^{N} (x_i - \mathbb{E}\{x\})^2}\sqrt{\sum_{i=1}^{N} (y_i - \mathbb{E}\{y\})^2}} \tag{15}$$

where x_i and y_i are the grayscale values of two selected adjacent pixels, N is the total number of pairs (x_i, y_i) obtained from the image, and $\mathbb{E}\{.\}$ denotes the expected value of a random variable. Table 3 reports the mean of absolute values of the correlation coefficients (horizontally, vertically, and diagonally) for 3000 randomly selected pairs of adjacent pixels from the original images and their encrypted ones for the

TABLE 3. Mean of absolute values of the correlation between pairs of the original and encrypted images.

	Scan direction	Original image	Encrypted image
Mean	Horizontal	0.9370	0.0025
	Vertical	0.8949	0.0029
	Diagonal	0.9009	0.0027

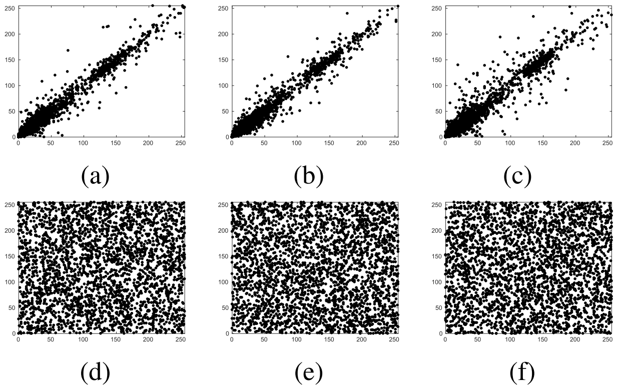


FIGURE 4. Distribution of pairs of adjacent pixels in the original and encrypted images of MR Shoulder. Frames (a) and (d): Distributions of two horizontally adjacent pixels in the original and encrypted images, respectively. Frames (b) and (e): Distributions of two vertically adjacent pixels in the original and encrypted images, respectively.
Frames (c) and (f): Distributions of two diagonally adjacent pixels in the original and encrypted images, respectively.

proposed algorithm. As can be seen from the table, the correlation coefficients of the encrypted images are approximately 0. The adjacent pixels in the encrypted image are now uncorrelated in the horizontal, vertical, and diagonal directions. This property is shown graphically in Figs. 4(a)-(c) and Figs. 4(d)-(f), which plot the correlations in the horizontal, vertical, and diagonal directions of the original and encrypted images of MR Shoulder, respectively. Therefore, the proposed encryption approach satisfies the zero correlation requirement and is highly resistant to correlation-based attacks.

3) HISTOGRAM AND CHI-SQUARE TEST

A histogram shows the gray level intensity of an image. This information can be very useful for histogram attacks against a non-uniform distribution. The images generated by a good encryption algorithm should have uniform histograms, so as to improve their resistance to statistical analysis. We have analyzed the histograms of many original images as well as their encryptions using the proposed approach. The original images of Lena, CR Chest and XA Coronary are shown in Fig. 5 along with their histograms, while Fig. 6 shows their encrypted counterparts alongside their histograms. Clearly, the histograms of the encrypted images are nearly uniform and exhibit significant differences from those of their corresponding original images. This uniformity is ensured by applying the Chi-square test [18] defined by the following equation

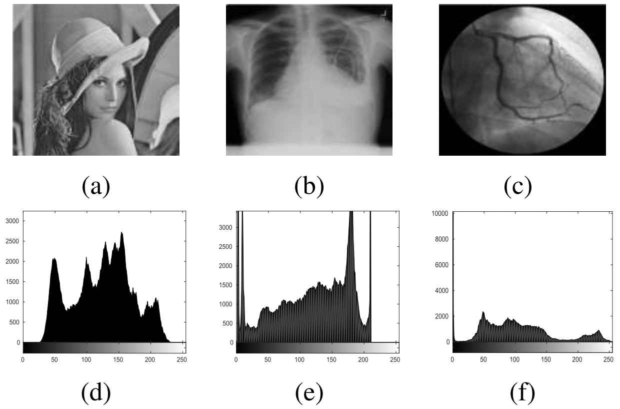


FIGURE 5. Original images and their histograms. Frames (a), (b) and (c) are the original images of Lena, CR Chest, and XA Coronary, respectively. Frames (d), (e) and (f) are the histograms of the original images of Lena, CR Chest, and XA Coronary, respectively.

TABLE 4. Chi-square test of histograms.

	χ^2 test		
Image	Score	p-value	Decision
CR Chest	239.4258	0.7500	H_0 accepted
CT Abdomen	243.7246	0.6834	H_0 accepted
CT Cardiac	226.2363	0.9024	H_0 accepted
CT Dental	249.7383	0.5812	H_0 accepted
CT Head	271.5176	0.2280	H_0 accepted
CT Paranasal sinus	213.1602	0.9735	H_0 accepted
MR Shoulder	262.8438	0.3545	H_0 accepted
XA Coronary	246.3828	0.6392	H_0 accepted

$$\chi^2 = \sum_{i=0}^{L-1} \frac{(o_i - e_i)^2}{e_i} \tag{16}$$

where L is the number of pixel levels, o_i is the observed occurrence frequency of each pixel value $(0 - 255)$ in the histogram of the encrypted image, and e_i is the expected frequency count of the uniform distribution, i.e., $e_i = (M \times N)/256$. The distribution is considered uniform. That is, the null hypothesis is accepted, when the p-value is more than the significance level s $(s \in [0, 1])$. Here, the p-value is the probability of observing a sample statistic as extreme as the χ^2 test. Table 4 presents the Chi-square scores and their p-values for the histograms of the encrypted images of size 512×512 and a significance level of 0.05. The obtained scores are smaller than $\chi^2_{th}(255, 0.05) = 293.247$, while their p-values are larger than 0.05. Therefore, the null hypothesis is accepted, namely, the histograms of the encrypted images are uniform. Consequently, it is concluded that the proposed encryption method is robust against statistical attacks based on the histogram analysis.

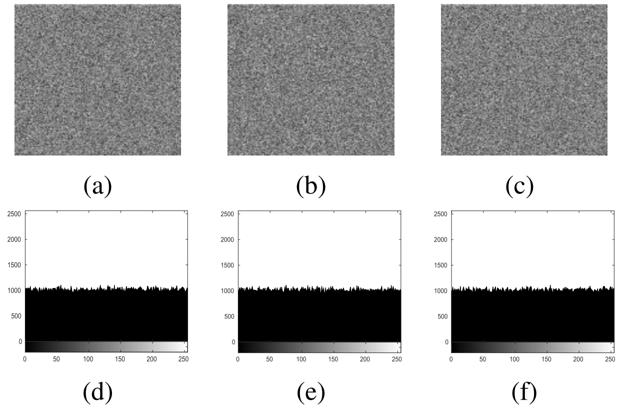


FIGURE 6. Encrypted images and their histograms. Frames (a), (b) and (c) are the encrypted images of Lena, CR Chest, and XA Coronary, respectively. Frames (d), (e) and (f) are the histograms of the encrypted images of Lena, CR Chest, and XA Coronary, respectively.

4) GLOBAL ENTROPY

Global entropy is a statistical measure of randomness, which is defined as

$$H(X) = -\sum_{k} p(x_k) \log_2 (p(x_k)) \quad \text{[bits]}, \tag{17}$$

where $p(x_k)$ is the probability of appearance of symbol x_k. The ideal entropy is obtained if all the pixels appear with the same probability, which means that the distribution of the pixels is uniform. The maximum or ideal entropy is equal to $\log_2(2^8) = 8$ bits. The mean and variance of the entropies of the different algorithms are given in Table 5. The mean entropy attainable by the proposed approach is very close to 8, approximately equal to that in [16], and slightly higher than those in [14] and [20]-BX.[1] This offers evidence that the proposed encryption approach is resistant entropy attacks.

TABLE 5. Global entropy (mean and variance) analysis.

	Global entropy			
	[16]	**[14]**	**[20]-BX**	**Proposed**
Mean	7.999311	7.999297	7.999306	**7.999324**
Variance $\left(\times 10^{-9}\right)$	3.872067	5.650593	5.148971	3.587629
Best for	6/16	0/16	3/16	**7/16**

5) LOCAL ENTROPY

Unlike the intrinsically quantitative global Shannon entropy, in which assessing the true randomness of an image [41] can fail, the local Shannon entropy is used to qualitatively measure the randomness of an image. It is defined as follows [41]. Let $\left(n, n_p\right)$ be the local Shannon entropy for the local image blocks using the following method:

- Select arbitrarily non-overlapping image blocks $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_n$ with n_p pixels inside a test image $\mathbf{B}$ with K intensity scales;
- Calculate the Shannon entropy $H\left(\mathbf{B}_j\right)$ using (17) for all $j \in \{1, 2, \ldots, n\}$; and
- Compute the sample average of the Shannon entropy of these n image blocks as follows

$$\overline{H_{n,n_p}}(\mathbf{B}) = \frac{1}{n} \sum_{j=1}^{n} H\left(\mathbf{B}_j\right). \tag{18}$$

The local Shannon entropies are compared in Table 6. As can be observed from the table, the proposed approach can achieve an average local entropy of 7.902512, which is very close to the ideal value of 7.9024693 [41]. Moreover, the proposed approach is able to achieve better average local entropy compared to other tested counterparts.

TABLE 6. Local entropy (mean and variance) analysis.

	Local entropy $(n = 30,\ n_p^{K=256^*} = 1936)$			
	[16]	**[14]**	**[20]-BX**	**Proposed**
Mean	7.902694	7.902396	7.902377	**7.902512**
Variance $\left(\times 10^{-7}\right)$	3.216059	2.996488	5.699458	3.662985
Best for	0/16	**6/16**	5/16	5/16

C. SENSITIVITY ANALYSIS

1) KEY SENSITIVITY

Key sensitivity is the number/quantity of the cipher image changes caused by a minor change applied to the secret key. For a good cipher, a small difference in the key should lead to significant modifications in the cipher image. Denote by $\mathbf{P} = p_0, p_1, \ldots, p_{MN-1}$, $\mathbf{C}^1 = c_0^1, c_1^1, \ldots, c_{MN-1}^1$, $\mathbf{C}^2 = c_0^2, c_1^2, \ldots, c_{MN-1}^2$, $\mathbf{key}^1 = key_0^1, key_1^1, \ldots, key_{MN-1}^1$ and $\mathbf{key}^2 = key_0^2, key_1^2, \ldots, key_{MN-1}^2$ the original image, the cipher image and the key, respectively. Hence, the key sensitivity K_s can be computed as follows

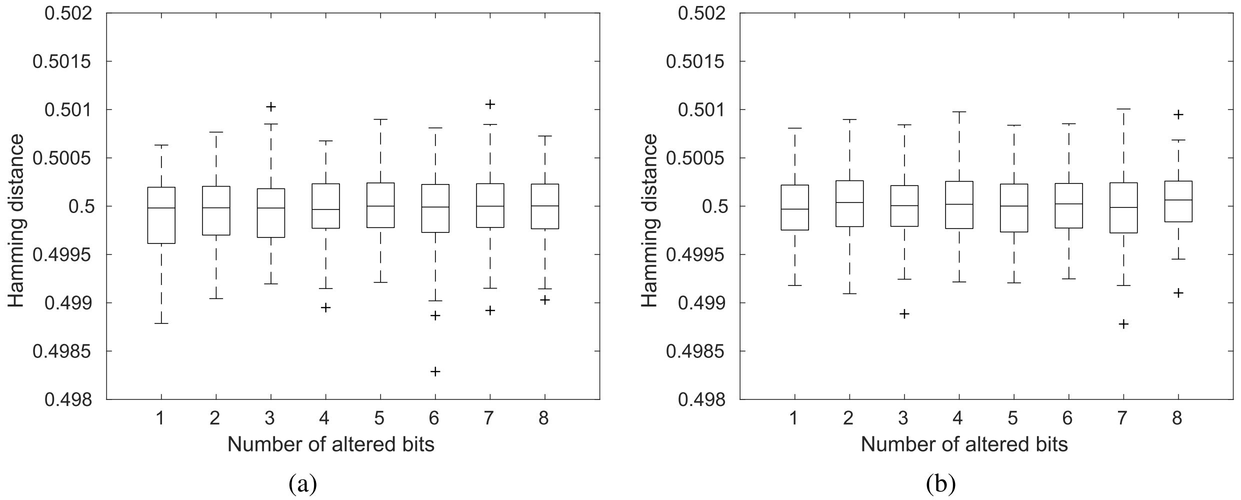
$$K_s = \frac{1}{MN} \sum_{j=0}^{MN-1} \left(c_j^1 \oplus c_j^2\right) \tag{19}$$

where $\mathbf{C}^1$ and $\mathbf{C}^2$ are given by

$$\begin{cases} \mathbf{C}^1 = \texttt{Encrypt}(\mathbf{P}, \mathbf{key}^1) \\ \mathbf{C}^2 = \texttt{Encrypt}(\mathbf{P}, \mathbf{key}^2). \end{cases}$$

Here, there are n bits of difference between $\mathbf{key}^1$ and $\mathbf{key}^2$, and `Encrypt(.)` denotes the encryption algorithm. Generally, the value of K_s for a good cipher is about 0.5. Fig. 7(a) shows the results of the key sensitivity test for the proposed approach. Under this test, the indices of the n $(n = 1, \ldots, 8)$ altered bits of the secret keys $\big($cf. (1) and (2)$\big)$ as well as the 512×512 images are randomly generated for 200 iterations. We can observe that, on average, 99.3% of the obtained K_s values are in [0.4991, 0.5008]. That is, they are very close to the optimal value of 0.5. Therefore, the proposed approach is highly sensitive to small changes within the encryption key.

[1] It stands for the BX version of the encryption scheme proposed in [20], where the pixel diffusion step is based on bitwise XOR (BX) operations.



(a)

(b)

FIGURE 7. Boxplot depicting the key sensitivity (a), and the plaintext sensitivity (b), where $\alpha_0 = 2.62890625$, $x_0 = 0.4296875$, $\alpha_1 = 3.7890625$, and $x_1 = 0.21484375$.

2) PLAINTEXT SENSITIVITY

Similar to key sensitivity, plaintext sensitivity is defined as the amount of changes in the ciphertext caused by a minor modification applied to the plaintext. For a good cipher, a small difference in the plaintext should cause considerable changes in the corresponding ciphertext. Let $\mathbf{P}^1 = p_0^1, p_1^1, \ldots, p_{MN-1}^1$, $\mathbf{P}^2 = p_0^2, p_1^2, \ldots, p_{MN-1}^2$, $\mathbf{C}^1 = c_0^1, c_1^1, \ldots, c_{MN-1}^1$, $\mathbf{C}^2 = c_0^2, c_1^2, \ldots, c_{MN-1}^2$ and $\mathbf{key} = key_0, key_1, \ldots, key_{MN-1}$ be the plaintext, the ciphertext and the key, respectively. Then the plaintext sensitivity P_s can be computed as follows

$$P_s = \frac{1}{MN} \sum_{j=0}^{MN-1} \left(c_j^1 \oplus c_j^2 \right) \tag{20}$$

where $\mathbf{C}^1$ and $\mathbf{C}^2$ are given by

$$\begin{cases} \mathbf{C}^1 = \texttt{Encrypt}(\mathbf{P}^1, \mathbf{key}) \\ \mathbf{C}^2 = \texttt{Encrypt}(\mathbf{P}^2, \mathbf{key}). \end{cases}$$

Here, there are n bits of difference between $\mathbf{P}^1$ and $\mathbf{P}^2$, and `Encrypt(.)` denotes the encryption algorithm. Generally speaking, for a good cipher, the value of P_s is about 0.5. Fig. 7(b) shows the plaintext sensitivity test for the proposed approach. Under this test, the indices of the n ($n = 1, \ldots, 8$) altered bits as well as the 512 × 512 images are randomly generated for 200 iterations. It is observed that, on average, 99.3% of the obtained P_s values fall within the range of [0.4992, 0.5009]. That is, the P_s values are always quite close to the optimal value of 0.5. Therefore, the proposed approach is extremely sensitive to small changes in the plaintext.

3) UACI AND NPCR

In a good encryption approach, the encrypted image is highly sensitive to slight changes in the original image (e.g., a single pixel change). In this context, two common quantitative measures are employed, namely NPCR (number of pixel change rate) and UACI (unified average changing intensity). We employ the NPCR to measure the number of different pixels. However, the average intensity difference is given by the UACI. The NPCR and UACI are defined by (21) and (22), respectively

$$\texttt{NPCR} = \frac{\sum_{i,j} \Phi(i,j)}{MN} \times 100\% \tag{21}$$

where $\Phi(i,j)$ is given by

$$\Phi(i,j) = \begin{cases} 0, & \text{if } \mathbf{F}_1(i,j) = \mathbf{F}_2(i,j) \\ 1, & \text{if } \mathbf{F}_1(i,j) \neq \mathbf{F}_2(i,j) \end{cases}$$

$$\texttt{UACI} = \frac{\sum_{i,j} |\mathbf{F}_1(i,j) - \mathbf{F}_2(i,j)|}{MN(2^n - 1)} \times 100\% \tag{22}$$

where $\mathbf{F}_1$ and $\mathbf{F}_2$ are two encrypted images whose corresponding original images have only a single-pixel difference, the $(i,j)th$ pixel of $\mathbf{F}_1$ and $\mathbf{F}_2$ are denoted by $\mathbf{F}_1(i,j)$ and $\mathbf{F}_2(i,j)$, respectively; n is the number of bits used to represent a grayscale pixel value.

To quantify the impact on the encrypted image of a one-bit pixel change in the original image, we calculated the NPCR and UACI for many grayscale images with the proposed and its comparative algorithms. The means and the variances of the NPCR and UACI for each algorithm are reported in Table 7.

As can be seen from Table 7, for the proposed algorithm, the mean NPCR and mean UACI values exceed 99.6% and 33.4%, respectively. Therefore, the proposed algorithm is highly sensitive to small pixel changes in the original image. In other words, the proposed encryption approach can make two encrypted images completely different, even given a single-bit difference between two source images. In addition, the proposed approach achieves better average NPCR and UACI performances compared to the others tested. As a result, the encryption approach is secure against differential attacks.

TABLE 7. NPCR and UACI (mean and variance) of the encrypted images due to one-bit change in the original image.

	NPCR (%)				UACI (%)			
	[16]	[14]	[20]-BX	Proposed	[16]	[14]	[20]-BX	Proposed
Mean	99.614 043	99.608 020	99.607 544	**99.617 340**	33.461 621	33.468 186	33.468 853	**33.475 593**
Variance	0.000 143	0.000 155	0.000 305	0.000 009	0.002 401	0.003 024	0.002 327	0.001 671
Best for	4/16	2/16	3/16	**7/16**	3/16	4/16	3/16	**6/16**

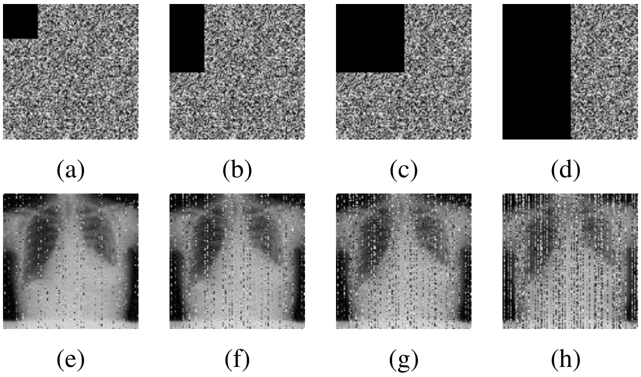


FIGURE 8. Test of image occlusion attacks: encrypted images with (a) 1/16, (b) 1/8, (c) 1/4, and (d) 1/2 data loss; corresponding decrypted images (e)-(h) in accordance with (a)-(d).

TABLE 8. PSNR between original and decrypted images subject to occlusion attacks.

		Occlusion			
	Algorithm	1/16	1/8	1/4	1/2
PSNR (dB)	[16]	12.0881	10.0969	8.8968	8.5539
	[14]	8.5675	8.5540	8.5502	8.5480
	[20]-BX	17.7218	14.8610	12.1275	9.7698
	Proposed	**20.6301**	**17.6447**	**14.6193**	**11.6147**

D. ROBUSTNESS ANALYSIS

1) OCCLUSION ATTACK

Encrypted images transmitted across a communications channel may be affected by data losses, which makes the decryption process incomplete or invalid. In this context, a loss operation known as the occlusion attack is applied on the encrypted images to test the ability of the proposed scheme in recovering the original image. The encrypted images of CR Chest with 1/16, 1/8, 1/4, and 1/2 data loss rates are shown in Figs. 8(a)-(d), respectively. The corresponding decrypted images from Figs. 8(a)-(d) are displayed in Figs. 8(e)-(h), respectively. It is observed that even if only half of an encrypted image has left unchanged, the corresponding decrypted image retains most of the original visual information. This demonstrates the efficacy of the proposed encryption algorithm to withstand occlusion attacks. Besides, we calculated the PSNR between the original image and the decrypted images for the proposed and its aforementioned comparative algorithms. The obtained results are listed in Table 8, where the proposed algorithm outperforms the comparative ones in resisting occlusion attacks.

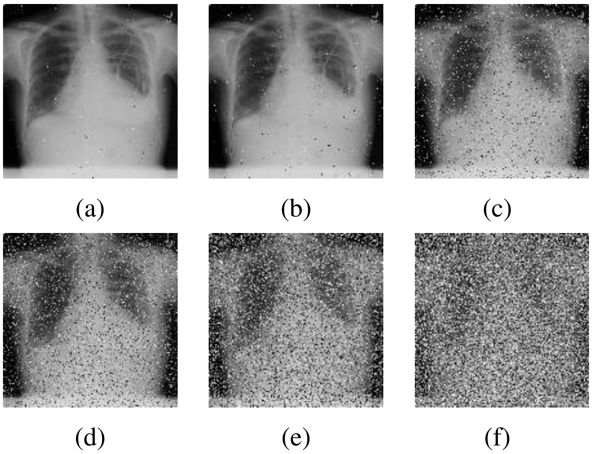


FIGURE 9. Decrypted images subject to the salt & pepper noise with a noise density of (a) 0.002; (b) 0.005; (c) 0.050; (d) 0.100; (e) 0.250; and (f) 0.500.

2) NOISE ATTACK

In real-world applications, digital images are transmitted through a communications channel which is usually subject to channel noise. To analyze the noise immunity of the proposed encryption algorithm, we assume that the encrypted image of CR Chest is contaminated by the salt & pepper noise of densities 0.0020, 0.0050, 0.0500, 0.1000, 0.2500, and 0.5000. The corresponding decrypted images are shown in Figs. 9(a)-(f), respectively. As can be seen from these figures, the decrypted images are noisy but perceivable. Moreover, the PSNR values between the decrypted and original images are listed in Table 9, which also includes a comparison with some related works. The given results demonstrate that the proposed algorithm outperforms the comparative ones in resisting noise attacks.

3) HISTOGRAM EQUALIZATION

Histogram equalization is a technique of strengthening image contrast using an image's histogram. Through this adjustment, the image intensity distribution becomes quite uniform. Therefore, the areas of lower local contrast benefit from higher contrast and the most redundant intensity values are efficiently spread out. The encrypted CR Chest image attacked by histogram equalization is displayed in Fig. 10(a). The corresponding decrypted image shown in Fig. 10(b) is blurry but recognizable. Under this attack, the PSNR value between the original and decrypted images using the proposed algorithm is 30.0205 dB, which exceeds those of

TABLE 9. PSNR between the original and decrypted images under noise.

		Density of salt & pepper noise					
	Algorithm	0.0020	0.0050	0.0500	0.1000	0.2500	0.5000
PSNR (dB)	**[16]**	26.1682	21.9976	12.8812	10.6900	8.8973	8.5504
	[14]	8.5900	8.5625	8.5514	8.5476	8.5454	8.5428
	[20]-BX	29.8380	25.6571	15.8923	13.1335	10.2166	8.8271
	Proposed	**32.8396**	**28.7068**	**18.8395**	**15.8599**	**12.2262**	**9.8903**

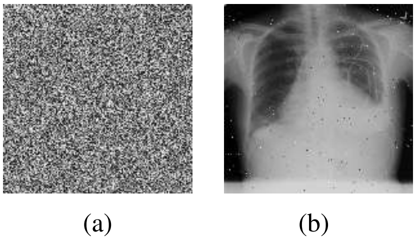

(a) (b)

FIGURE 10. Test of histogram equalization attack: (a) encrypted image under histogram equalization attack; and (b) its decrypted image.

TABLE 10. PSNR values between the original and decrypted images under histogram equalization attacks.

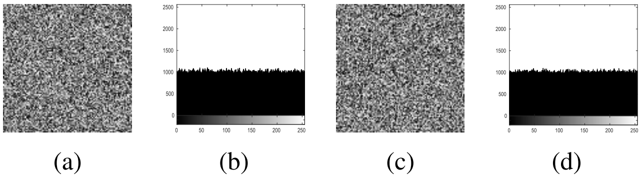
	[16]	[14]	[20]-BX	Proposed
PSNR (dB)	23.7657	8.5605	26.1328	**30.0205**

the comparative algorithms (see Table 10). Consequently, the proposed algorithm can well resist histogram equalization attacks.

E. KNOWN-PLAINTEXT AND CHOSEN-PLAINTEXT ATTACKS

Following the sufficiently recognized principle of Kerckhoffs' in the cryptology community [42], we assume that all details of the encryption/decryption cryptosystems are known to attackers, which implies that the security of an algorithm depends only on the decryption key. Therefore, the main task of cryptanalysis is to restructure the key or its equivalent form, which can effectively decrypt all or partial contents of any encrypted plaintext by the cipher. Furthermore, a cryptanalyst can find a method to retrieve the plaintext from the ciphertext without identifying the secret key. In fact, he/she looks for correlations between the ciphertext and the key or the ciphertext and the plaintext. This search can be carried out based upon different assumptions leading to four different attacks listed below in descending order (from the hardest to the easiest attacks), namely ciphertext-only attack, known plaintext attack, chosen plaintext attack, and chosen ciphertext attack. Among them, the known-plaintext and chosen-plaintext attacks are the stronger ones, and if an algorithm can resist them, it can hold off the other attacks. In the proposed algorithm, the SHA-256 hash function is employed to compute the initial values and the control parameters of the chaotic maps. Consequently, they are susceptible to a single bit difference in the secret key or the original image. Thus, they are correlated with both the secret key and the original image. This brings about the generation of dissimilar chaotic sequences for different original images even with the same collection of keys. Consequently, a cryptanalyst who tries to analyze the proposed algorithm cannot obtain useful information about both the secret key and the output image, since these information depends on the input image. Accordingly, the proposed algorithm can resist the known-plaintext and chosen-plaintext attacks.

In an attack scenario, attackers always employ all white and all black images to make the permutation/substitution processes of the ciphers invalid, and then try to infer some useful information. The statistical and differential analyses of the all white and all black images are shown in Fig. 11 and Table 11. The obtained encrypted images are random-like images, so no useful information can be extracted from them. The histogram distributions shown in Figs. 11(b) and 11(d) are quite uniform, the correlation values in all directions are close to 0, the global entropy is nearly equal to 8, and the local entropy is very close to the ideal value of 7.9024693. Moreover, the obtained NPCR and UACI values are quite close to the expected ones, i.e., 99.6% and 33.4%, respectively, which proves the effectiveness of the proposed algorithm against differential attacks. With regards to the experimental results, the proposed algorithm can effectively encipher the all white and all black images and resist known-plaintext and chosen-plaintext attacks.



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

FIGURE 11. Encrypted images of the all white (a) and all black (c); their histograms (b) and (d), respectively.

F. FURTHER COMPARISON WITH SOME RECENT DNA-BASED ALGORITHMS

To compare the performances of the proposed algorithm and demonstrate its superiority with regards to existing DNA-based ones [24]–[30], we conduct a comparative study based upon some performance indicators as in Table 12. The 8-bit Lena image of size 256 × 256 is used as an original image, and the results regarding the key space, correlation, entropy, robustness, UACI, and NPCR are presented. The results for the algorithms [25]–[29] are directly reported

TABLE 11. Statistical and differential analyses of the encrypted ones of the all white and all black images.

Image	Histogram (χ^2 test)		Correlation			Global entropy	Local entropy	NPCR (%)	UACI (%)
	Score	p-value	Horizontal	Vertical	Diagonal				
All white	262.0176	0.3679	0.0038	−0.0040	−0.0005	7.9993	7.9017	99.6300	33.4748
All black	230.6504	0.8610	0.0027	−0.0004	0.0053	7.9994	7.9019	99.5968	33.4218

TABLE 12. Comparison results between the proposed and some DNA-based algorithms. The original image is Lena 256 × 256.

Algorithm	Key space	Correlation			Global entropy	Robustness analysis	NPCR (%)	UACI (%)
		Horizontal	Vertical	Diagonal				
[24]	$\approx 2^{298}$	0.0068	−0.0054	0.0010	7.9967	No	99.6100	33.4600
[25]	$\approx 2^{233}$	−0.0013	0.0074	−0.0051	7.9969	No	0.0046	0.0009
[26]	$\approx 2^{295}$	0.0214	0.0465	−0.0090	7.9993	No	99.6000	33.4400
[27]	$\approx 2^{182}$	0.0011	−0.0071	−0.0005	7.9970	No	99.5911	33.4798
[28]	$\approx 2^{299}$	−0.0006	−0.0083	0.0056	7.9897	Yes	99.6033	33.4655
[29]	$\approx 2^{372}$	−0.0049	0.0142	−0.0155	7.9967	No	99.6201	33.3551
[30]	2^{128}	−0.0016	−0.0033	0.0130	7.9971	Yes	99.6200	33.4500
Proposed	$\geq 2^{716}$	0.0013	−0.0049	0.0057	7.9974	Yes	99.6536	33.4121

from [43]. The proposed scheme possesses the largest key space, and it is the second best algorithm concerning entropy, while the algorithm proposed in [26] is ranked first. The studied algorithms have similar correlation coefficients that reach nearly null correlation. Only the proposed algorithm and those in [28] and [30] are proven robust against occlusion attacks, noise attacks, and histogram equalization attacks. Table 12 shows that the NPCR and UACI values of the studied algorithms are all close to the expected values, i.e., 99.6% and 33.4% respectively, except for the algorithm proposed in [25], where both the NPCR and UACI values are approximately equal to zero and the algorithm is vulnerable to differential attacks. It is clear that the proposed scheme exceeds most of the existing ones concerning aforementioned criterions. Indeed, it has stronger resistance to brute force attacks than the other comparative algorithms, while the proposed algorithm and those in [28] and [30] are immune to statistical and differential attacks.

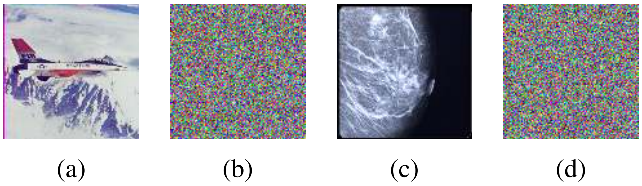


FIGURE 12. Original images and their encrypted ones: (a), (c) original images of Airplane and Mammogram, respectively; corresponding encrypted images (b), (d) from (a), (c).

G. EXTENSION OF THE PROPOSED ALGORITHM TO COLOR IMAGES

The proposed algorithm can be extended to encrypt RGB images. A color image is first decomposed into three components of R, G, and B, each of which is then encrypted using the proposed algorithm. After that, the encrypted components are merged, yielding the encrypted RGB image. The RGB original images of Airplane and Mammogram are shown in Figs. 12(a) and 12(b), respectively. Their corresponding encrypted images are illustrated in Figs. 12(c) and 12(d), respectively. These results demonstrate that our proposed algorithm is suitable for color images.

H. TIME COMPLEXITY ANALYSIS

TABLE 13. Time complexity orders and their orders of magnitude for a grayscale image of size $M \times N = 512 \times 512$.

Algorithm	Complexity order	Order of magnitude
[16]	$O\left(MN\left(34+\log(L)+2\sqrt{L}\right)\right)$	282×10^6
[20]-BX	$O\left(40MN\right)$	10×10^6
[14]	$O\left(108MN+72L^4\right)$	45×10^6
[24]	$O\left(579MN\right)$	151×10^6
Proposed	$O\left(124MN\right)$	32×10^6

Table 13 shows the time complexity orders and their orders of magnitude for a 512 × 512 grayscale image using the proposed algorithm compared to its counterparts in [16] and [20]-BX, [14], and [24]. This table is derived from a complexity analysis detailed in Appendix. Through this analysis, the proposed algorithm is shown to be the second faster one after the one in [20]-BX. This confirms that our proposed algorithm has a good time complexity.

V. CONCLUSION

This paper proposed a new chaos-based encryption scheme for medical images. The cryptosystem consists of two iterative phases, preceded by a key generation layer that employs the SHA-256 hash function. Each phase of the encyption scheme is composed of block-based permutation, pixel-based substitution, DNA encoding, bit-level substitution, DNA decoding, and bit-level diffusion. The key-streams in the bit-level substitution are generated by the logistic-Chebyshev map, while the sine-Chebyshev map is employed to produce

TABLE 14. Time complexity analysis for the evaluated image encryption algorithms.

	[16](1)	[20]-BX	[14](5)	[24]	Proposed
Addition	$18MN+260$	$18MN+20M+20N+306$	$44MN+36L^4+8L^2+32$	$132MN+39$	$28MN+70$
Multiplication	$10MN+266$	$10MN+27M+27N+324$	$40MN+16L^4+8L^2+36$	$150MN+3$	$52MN+2v+24$
Exponentiation	259	256	128	$8MN$	–
Absolute value	–	–	–	$135MN$	–
Trigonometric functions	–	$2MN+6M+6N+16$	–	–	$22MN$
Mod	$4MN+6$	$6MN+16M+16N+40$	$16MN+12L^4+8L^2+32$	$68MN+13$	$12MN+2v+2$
Rounding functions	$2MN$	$2M+2N+4$	$8MN+8L^4+8L^2+34$	$67MN+2$	–
XOR	–	$4MN+8M+8N+16$	16	$6MN$	$2MN+90$
Substitutions	–	–	–	$13MN$	$8MN$
Sorting vector of length	$\sqrt{L}$(2)	$M+2$,(4) $N+2$(4)	L^2(6), L^2(6)	–	–
Searching in vector of length	$\sqrt{L}$(3)	–	–	–	–
SHA-256 operations	–	–	–	–	$\frac{8}{512}MN+1$

(1) The data block is of length L. Then, each block is assumed to be of size $\sqrt{L}\times\sqrt{L}$. For the given simulation results in [16], L is equal to MN;

(2) One sorting operation for each round, data block, and block row. The sorting-time complexity should be multiplied by $2\times\frac{MN}{L}\times\sqrt{L}=\frac{2MN}{\sqrt{L}}$;

(3) One searching operation for each round, data block, block row, and block column. The searching-time complexity should be multiplied by $2\times\frac{MN}{L}\times\sqrt{L}\times\sqrt{L}=2MN$;

(4) The sorting-time complexity should be multiplied by 2, i.e., the number of rounds;

(5) The data block is of size $L\times L$. In [14], $L=\min\{\lfloor\sqrt{M}\rfloor,\lfloor\sqrt{N}\rfloor\}$, where $\lfloor.\rfloor$ denotes the floor function;

(6) The sorting-time complexity should be multiplied by 4, i.e., the number of rounds.

the key-streams in the bit-level diffusion. Several security and performance analyses were carried out to validate the effectiveness of the proposed scheme. It was shown that the key space of the proposed scheme is larger than 2^{716}, and the average correlation between pairs of encrypted images is 0.0008. The average global and local entropy are 7.9993 and 7.9025, respectively. Moreover, whether for the original image or the encryption key, the test of sensitivity to bit alteration gives very close values to the optimum. Additionally, the average values of the NPCR and UACI are 99.6179% and 33.4784%, respectively. The time complexity of the proposed algorithm is lower compared to some recently reported image encryption algorithms. Considering the above results, the proposed scheme fulfills all the necessary criteria of a good encryption scheme, including a large key space, resistance to statistical attacks, differential, chosen/known plaintext attacks. Moreover, the proposed algorithm has linear running time, implying that it is computationally efficient.

APPENDIX
TIME COMPLEXITY ANALYSIS FOR THE PROPOSED IMAGE ENCRYPTION ALGORITHM AS WELL AS THOSE IN [16], [20]-BX, [14], AND [24]

Table 14 describes how many times each operation is repeated according to the number of pixels MN. With regards to searching and sorting operations, considering a vector of n elements, searching an element in the vector is of time complexity $O(n)$. Assuming the use of the *quicksort* algorithm [44], sorting the vector is of average time complexity $n\log(n)$. The SHA-256 hash algorithm entry is of size 512 bits, and the algorithm has a constant number of operations. The algorithms in [14] and [20] use randomly generated numbers. To derive their time complexities, we consider a linear congruential generator since it is widely used as a uniform random number generator [45].

After gathering the operations given by Table 14, and ignoring lower-orders terms, the time complexities can be shown as

$$\begin{cases} O\left(MN\left(34+\log(L)+2\sqrt{L}\right)\right) & \text{for [16]} \\ O\left(40MN\right) & \text{for [20]-BX} \\ O\left(108MN+72L^4\right) & \text{for [14]} \\ O\left(579MN\right) & \text{for [24]} \\ O\left(124MN\right) & \text{for our proposed algorithm.} \end{cases} \tag{23}$$

REFERENCES

[1] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 243–255, Feb. 2015.

[2] S. Ahani and S. Ghaemmaghami, "Colour image steganography method based on sparse representation," *IET Image Process.*, vol. 9, no. 6, pp. 496–505, Mar. 2015.

[3] N. M. Makbol, B. E. Khoo, T. H. Rassem, and K. Loukhaoukha, "A new reliable optimized image watermarking scheme based on the integer wavelet transform and singular value decomposition for copyright protection," *Inf. Sci.*, vol. 417, pp. 381–400, Nov. 2017.

[4] T. Zong, Y. G. Xiang, S. Guo, and Y. Rong, "Rank-based image watermarking method with high embedding capacity and robustness," *IEEE Access*, vol. 4, pp. 1689–1699, 2016.

[5] G. Ye and X. Huang, "An image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia*, vol. 23, no. 2, pp. 64–71, Apr./Jun. 2016.

[6] M. Dridi, M. A. Hajjaji, B. Bouallegue, and A. Mtibaa, "Cryptography of medical images based on a combination between chaotic and neural network," *IET Image Process.*, vol. 10, no. 11, pp. 830–839, 2016.

[7] A. Al-Haj, G. Abandah, and N. Hussein, "Crypto-based algorithms for secured medical image transmission," *IET Inf. Secur.*, vol. 9, no. 6, pp. 365–373, May 2015.

[8] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[9] W. Cao, Y. Zhou, C. L. P. Chen, and L. Xia, "Medical image encryption using edge maps," *Signal Process.*, vol. 132, pp. 96–109, Mar. 2017.

[10] J. B. Lima, F. Madeiro, and F. J. R. Sales, "Encryption of medical images based on the cosine number transform," *Signal Process., Image Commun.*, vol. 35, pp. 1–8, Jul. 2015.

[11] D. Levy, "Chaos theory and strategy: Theory, application, and managerial implications," *Strategic Manage. J.*, vol. 15, no. S2, pp. 167–178, Jun. 1994.

[12] A. Belazi, R. Rhouma, and S. Belghith, "A novel approach to construct S-box based on Rossler system," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 611–615.

[13] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[14] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," *Inf. Sci.*, vol. 396, pp. 97–113, Aug. 2017.

[15] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, no. 11, pp. 172–182, 2014.

[16] Y. Zhou, Z. Hua, C. M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.

[17] A. Kanso and M. Ghebleh, "An efficient and robust image encryption scheme for medical applications," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 24, nos. 1–3, pp. 98–116, Jul. 2015.

[18] S. E. Assad and M. Farajallah, "A new chaos-based image encryption system," *Signal Process. Image Commun.*, vol. 41, pp. 144–157, Feb. 2016.

[19] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.

[20] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 144, pp. 134–144, Mar. 2018.

[21] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[22] G. Xiao, M. Lu, L. Qin, and X. Lai, "New field of cryptography: DNA cryptography," *Chin. Sci. Bull.*, vol. 51, no. 12, pp. 1413–1420, Jun. 2006.

[23] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, Jan. 2018.

[24] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–14, Apr. 2018.

[25] Q. Zhang, L. Guo, and X. Wei, "A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Optik*, vol. 124, no. 18, pp. 3596–3600, Sep. 2013.

[26] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools Appl.*, vol. 75, no. 11, pp. 6303–6319, 2016.

[27] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, Oct. 2015.

[28] X. Wu, H. Kan, and J. Kurths, "A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Appl. Soft Comput.*, vol. 37, pp. 24–39, Dec. 2015.

[29] X. Huang and G. Ye, "An image encryption algorithm based on hyper-chaos and DNA sequence," *Multimedia Tools Appl.*, vol. 72, no. 1, pp. 57–70, Sep. 2014.

[30] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Process., Image Commun.*, vol. 52, pp. 6–19, Mar. 2017.

[31] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 23, nos. 1–3, pp. 294–310, Jun. 2015.

[32] Y.-Q. Zhang and X.-Y. Wang, "Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation," *Nonlinear Dyn.*, vol. 77, no. 3, pp. 687–698, Mar. 2014.

[33] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.

[34] W. Zhang, K.-W. Wong, H. Yu, and Z.-L. Zhu, "A symmetric color image encryption algorithm using the intrinsic features of bit distributions," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 3, pp. 584–600, Mar. 2013.

[35] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, Jan. 2017.

[36] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "An efficient approach for the construction of LFT S-boxes using chaotic logistic map," *Nonlinear Dyn.*, vol. 71, nos. 1–2, pp. 133–140, Jan. 2013.

[37] I. I. Cisse, H. Kim, and T. Ha, "A rule of seven in Watson-Crick base-pairing of mismatched sequences," *Nature Struct. Mol. Biol.*, vol. 19, no. 6, pp. 623–627, Jun. 2012.

[38] H. Liu and X. Wang, "Triple-image encryption scheme based on one-time key stream generated by chaos and plain images," *J. Syst. Softw.*, vol. 86, no. 3, pp. 826–834, Mar. 2013.

[39] (2015). *The USC-SIPIb image database, last visited.* [Online]. Available: http://sipi.usc.edu/database/

[40] (2017). *Image Processing Place, last visited.* [Online]. Available: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm

[41] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci.*, vol. 222, pp. 323–342, Feb. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002002551200521X

[42] D. R. Stinson, *Cryptography: Theory and Practice*, 3rd ed. Boca Raton, FL, USA: CRC Press, Nov. 2005.

[43] T. Hu, Y. Liu, L.-H. Gong, S.-F. Guo, and H.-M. Yuan, "Chaotic image cryptosystem using DNA deletion and DNA insertion," *Signal Process.*, vol. 134, pp. 234–243, May 2017.

[44] C. A. R. Hoare, "Algorithm 64: Quicksort," *Commun. ACM*, vol. 4, no. 7, p. 321, Jul. 1961. doi: 10.1145/366622.366644.

[45] P. L'Ecuyer, "Uniform random number generation," *Ann. Oper. Res.*, vol. 53, no. 1, pp. 77–120, Dec. 1994.

AKRAM BELAZI received the engineering degree in telecommunication and networks from the National Engineering School of Gabes, ENIG Tunisia, in 2011, the M.Sc. degree in electronic systems and networks communications from the Polytechnic School of Tunisia, EPT, in 2013, and the Ph.D. degree in telecommunications from the National Engineering School of Tunis, ENIT Tunisia, in 2017. His current research interests include multimedia cryptography, machine learning, deep learning, and optimization techniques.

MUHAMMAD TALHA received the Ph.D. degree in computer science from the Faculty of Computing, Universiti Teknologi Malaysia. He is currently a Researcher with the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia. He has authored more than 25 international journals and conferences. His research interests include image processing, medical imaging, features extraction, classification and machine learning techniques.

SOFIANE KHARBECH (M'17) received the Engineering degree in networking and telecommunications from the National Institute of Applied Science and Technology, Tunis, Tunisia, in 2009, the M.Sc. degree in electronic systems and communication networks from the Tunisia Polytechnic School, Carthage, Tunisia, in 2012, and the Ph.D. degree in electrical engineering from the University of Valenciennes and Hainaut-Cambresis, Valenciennes, France, in 2015.

From 2012 to 2015, he was a Research Engineer with the Laboratory IEMN/DOAE (CNRS UMR 8520, France). Since 2015, he has been an Assistant Professor with the Higher Institute for Technological Studies of Gabes, Gabes, Tunisia, and a Senior Researcher with the Laboratory Sys'Com-ENIT, Tunisia. His current research interests include cognitive radio and wireless communications.

WEI XIANG (S'00–M'04–SM'10) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 1997 and 2000, respectively, and the Ph.D. degree in telecommunications engineering from the University of South Australia, Adelaide, Australia, in 2004. From 2004 to 2015, he was with the School of Mechanical and Electrical Engineering, University of Southern Queensland, Toowoomba, Australia. He is currently the Founding Professor and the Head of discipline of Internet of Things engineering with James Cook University, Cairns, Australia. Due to his instrumental leadership in establishing Australia's first accredited Internet of Things Engineering degree program, he was selected into Pearcy Foundation's Hall of Fame, in 2018. He has published more than 250 peer-reviewed papers with more than 130 journal articles. His research interests include communications and information theory, particularly the Internet of Things, and coding and signal processing for multimedia communications systems. He is an elected Fellow of the IET, U.K., and Engineers Australia. He was named a Queensland International Fellow by the Queensland Government of Australia, from 2010 to 2011, an Endeavour Research Fellow by the Commonwealth Government of Australia, from 2012 to 2013, a Smart Futures Fellow by the Queensland Government of Australia, from 2012 to 2015, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society for Promotion of Science, from 2014 to 2015. He received the TNQ Innovation Award, in 2016, the Pearcey Entrepreneurship Award, in 2017, and Engineers Australia Cairns Engineer of the Year, in 2017. He was a co-recipient of the four Best Paper Awards at 2009 ICWMC, 2011 IEEE WCNC, 2015 WCSP, and 2019 WiSATS. He has been awarded several prestigious fellowship titles. He is the Vice Chair of the IEEE Northern Australia Section. He was an Editor for the IEEE Communications Letters, from 2015 to 2017, and is an Associate Editor for Springer's *Telecommunications Systems*. He has severed in a large number of international conferences in the capacity of a General Co-Chair, a TPC Co-Chair, and a Symposium Chair.