Soft Computing (2022) 26:7461–7477
https://doi.org/10.1007/s00500-021-06459-w

FOCUS

A new color image encryption technique using DNA computing and Chaos-based substitution box

Fawad Masood[1,9] · Junaid Masood[2] · Lejun Zhang[1] · Sajjad Shaukat Jamal[3] · Wadii Boulila[4,5] · Sadaqat Ur Rehman[6] · Fadia Ali Khan[7,8] · Jawad Ahmad[9]

Accepted: 19 October 2021 / Published online: 1 December 2021

Abstract

In many cases, images contain sensitive information and patterns that require secure processing to avoid risk. It can be accessed by unauthorized users who can illegally exploit them to threaten the safety of people's life and property. Protecting the privacies of the images has quickly become one of the biggest obstacles that prevent further exploration of image data. In this paper, we propose a novel privacy-preserving scheme to protect sensitive information within images. The proposed approach combines deoxyribonucleic acid (DNA) sequencing code, Arnold transformation (AT), and a chaotic dynamical system to construct an initial S-box. Various tests have been conducted to validate the randomness of this newly constructed S-box. These tests include National Institute of Standards and Technology (NIST) analysis, histogram analysis (HA), nonlinearity analysis (NL), strict avalanche criterion (SAC), bit independence criterion (BIC), bit independence criterion strict avalanche criterion (BIC-SAC), bit independence criterion nonlinearity (BIC-NL), equiprobable input/output XOR distribution, and linear approximation probability (LP). The proposed scheme possesses higher security wit NL = 103.75, SAC $\approx$ 0.5 and LP = 0.1560. Other tests such as BIC-SAC and BIC-NL calculated values are 0.4960 and 112.35, respectively. The results show that the proposed scheme has a strong ability to resist many attacks. Furthermore, the achieved results are compared to existing state-of-the-art methods. The comparison results further demonstrate the effectiveness of the proposed algorithm.

Keywords Privacy-preserving visual recognition · Image encryption · DNA sequence · Arnold transform · Chaotic system · Nonlinear component · S-box

1 Introduction

The trend of transmitting digital information over the Internet is growing exponentially. While this increases convenience and accessibility, extra challenges also increase with every development and improvement in technology. One of the inevitable issues is providing adequate security for data transmission that utilizes insecure communication networks. The number of connected users grows every day, as does their diverse Internet activity. As a result, the numbers and types of potential cybersecurity assaults have increased as well. This creates further challenges because data is an organization's most essential asset in today's world. Protecting sensitive data from unauthorized access has become a critical priority because attackers may use open public Internet for exploitative or malicious purposes. To avoid such attacks, sensitive data requires modification into cipherable forms before being transmitted via unencrypted channels. Confidential information requires a speedy, reliable, and robust cryptosystem to prevent information leakage.

Both researchers and academics have been exploring multiple alternative approaches to protecting transmitted data. With recent developments in communication technology, many encryption algorithms are designed for the security of real-time communication. Cryptography also plays an essential role in providing security for sensitive information. A wide range of algorithms has been presented to this end, including advanced encryption standard (AES), data encryption standard (DES), Elliptic curve cryptography (ECC), and so on. Many attempts have also been made to break down specific algorithms based on

Communicated by Irfan Uddin.

Extended author information available on the last page of the article

advanced encryption standards (AES) and data encryption standards (DES), which have been successful in some instances since 1993.

Regardless of these outliers, cryptography is still one of the most effective methods for preserving sensitive data. With expanding growth of new Internet channels and technologies, more sophisticated cryptanalysis and more robust and efficient image encryption techniques have become necessary for secure data communication. This is because cryptography encodes and transmits data in a specific format that can only be read and processed by those authorized to use advanced mathematical concepts. Encryption, or the act of encoding a communication in a format that unauthorized users cannot read or understand, is a crucial part of cryptography. Encryption in its various forms has been used since the Romans and even earlier, but increasingly complex versions are needed to keep up with new needs. A plain text can be encrypted into ciphertext and then the data can be sent over an insecure transmission medium. Depending on the security of the algorithm, the ciphertext may not be accessed by an unauthorized person.

A variety of symmetric and asymmetric image cryptographic algorithms have also been developed. In symmetric key cryptography, for instance, both users (i.e., sender and receiver) use a single key for the process of ciphering and deciphering. By contrast, asymmetric key cryptography utilizes two keys, a public key and a secret one, at each point to achieve additional security. In this approach, the private key is always kept secure because it decrypts the information. In contrast, the public key is always made publicly available to everyone because it does not help us decrypt the secret information.

In addition, most modern encryption designs are based on chaotic systems. Symmetric key cryptographic algorithms are significant because they produce a strong key for these cryptosystems and are very cheap. The keys are considerably smaller for the degree of security they provide, and running these algorithms is relatively inexpensive. Chaotic maps have also garnered a great deal of attention over the past few decades as another means of protecting cryptographic algorithms. Chaotic cryptography can secure communication further in a shorter duration of time. Quantum image processing is another method that is also becoming more popular to ensure information confidentiality. However, there are multiple proposals for data encryption in the literature. Chaotic systems, quantum encryption, and substitution boxes (S-boxes) are all often used.

Several nonlinear methods have been proposed to combat cryptographic attacks. In past, many image encryption schemes are mainly based on a chaotic dynamical system. The behavior of dynamical systems are pseudorandom and hence suited for multimedia encryption. The output of chaos maps is based on initial conditions. For this reason, chaos-based systems are known as deterministic systems. Their nature of randomness, sensitivity to original conditions, and ergodicity are unique characteristics (Stallings 2006; Chuang et al. 2011; Al-Najjar 2012; Banthia and Tiwari 2013; Rivest 1990; Matthews 1989; Wheeler and Matthews 1991; Chen and Liao 2005; Masood et al. 2020a, 2021, 2020b; Ahmad et al. 2020; Hanouti et al. 2020; Butt et al. 2020; Munir et al. 2020). These characteristics lead to a reliable cryptosystem, while chaotic maps and dynamical systems help to generate long-term chaotic sequences. Here, even a small change in initial conditions will significantly shift the chaotic sequence initially developed. These properties make these options some of the best choices for constructing secure algorithms in cryptography. By contrast, many techniques based on cryptanalysis are offered as a means of securing cryptographic algorithms, in turn depicting weakness in existing cryptosystems (Munir et al. 2021a, 2021b; Hanouti et al. 2021a, 2021b).

DNA computing and its intrinsic properties have been used extensively in the field of cryptography. Massive parallelism, high-level computational capacity, and storing large amounts of data are among these inherent properties. Research in this area often utilizes publicly accessible biological data to encrypt plaintext data in DNA computing applications. Adleman (Adleman 1994, 1998; Jiao and Goutte 2008) was the first to propose cryptographic DNA computing in 1994, initiating a new era of data processing that provides DNA-based encryption algorithms with tangible advantages over conventional cryptographic techniques. However, encrypting images with DNA encoding alone are inefficient. As a result, the underlying vulnerability problems are often solved using encryption techniques utilizing DNA computing and chaotic sequences (Enayatifar et al. 2014; Naskar and Chaudhuri 2016; Hanouti and Fadili 2021). For example, Clelland et al. (1999) have developed an innovative approach to protect secret communications using human genomic DNA. Meanwhile, Xiuli et al. (Chai et al. 2017) created a unique encryption method by adding chaotic maps and DNA sequences. A matrix based on DNA is created initially, and then, a plaintext image is stored before the circulation permutation process of row and column-wise is added. Yueping et al. (Li et al. 2017) have also offered a secure cryptographic technique. These proposed cryptosystems take high-dimensional chaotic maps to get robust security. Yueping et. al's. systems could withstand various assaults based on chosen plain text and cipher text methods, and their proposed scheme works rapidly and efficiently.

Many other researchers (Mondal and Mandal 2017) have also developed effective and lightweight encryption schemes that use DNA and chaotic approaches. Here, the unencrypted image utilizes confusion with randomly generated numbers obtained from a chaotic logistic map (employed cross-linked). In one approach, the pixels are then distorted with the computational method of DNA. For instance, Chen et al. (2018) have presented a cryptosystem based on the pixel's permutation and distortion process, which works on the self-adaptive process and is an efficient method due to its randomized but reusable variables. The last stage takes the DNA encoding method. In the Rijndael cipher (Daemen and Rijmen 1998), the work of well-known Belgian cryptographers Vincent Rijmen and Joan Daemen was selected as the advanced encryption standard (AES) in October 2000. The S-box based on AES is often regarded as the highest benchmark in this field. The optimum highly nonlinear value is 120, and the most significant value obtained by the AES S-box is 112 (Rijndael).

Following this lead, many more S-boxes have been developed to provide even stronger alternatives. For example, S-boxes with cryptographic features, such as the AES S-box, can be employed. Our work proposes a novel method for designing a robust substitution box (S-box) with better cryptography features. This S-box helps to substitute original data into plain text while maintaining its entropy level. We used one-dimensional (1D) and two-dimensional (2D) chaotic maps and DNA sequencing to construct this S-box. The sequence generated is filtered to unique random elements of a 256 count. The entropy value of 8 approximates an ideal value that satisfies the complete randomness needed from our proposed S-box.

Following its construction, our new S-box is investigated using multiple randomness and performance analysis tests, whose results show that our constructed S-box is exceptional for implementing real-time communication. Today, most cryptographers work in advanced encryption standard (AES) because of its highly robust cryptographic algorithm. In modern cryptography, block encryption algorithms play an essential role in providing security, such as international data encryption standards (IDES) and advanced encryption standards (AES). Due to their prominent chaos features, S-boxes are a superior choice for designing cryptosystems. At the same time, several security tests of S-boxes support the proposed cryptosystem's strength against both differential and linear attacks (Sani et al. 2021; Azam et al. 2021; Qayyum et al. 2020; Zahid et al. 2021; Liu et al. 2021). Thus, S-boxes form one of the fundamental nonlinear components used to provide security for cryptographic schemes.

1.1 Contribution

The following are the key contribution of our research study:

- Presenting efficient cryptosystem that uses combined effect of DNA and chaotic dynamical system for the development of initial S-box.
- The system uses multiple stages that help to generate highly random sequencing that exhibit minimum correlation.
- The proposed system uses both substitution and permutation for an extra layer of security. Both substitution and permutation ensure higher image security.
- Investigation of various existing state-of-the-art methods and comparing them with the proposed scheme.
- The proposed scheme is investigated thoroughly using various tests, i.e., nonlinearity (NL), strict avalanche criterion (SAC), bit independence criterion (BIC), bit independence criterion strict avalanche criterion (BIC-SAC), bit independence criterion nonlinearity (BIC-NL), equiprobable input/output XOR distribution, and linear approximation probability.

2 Fundamental concepts

2.1 Arnold transformation (AT)

Shuffling the pixels of an initial image is one of the essential elements used to provide image security. Here, the security of an image can be accomplished by applying this one image transformation method. There are various shuffling methods; however, Arnold transformation (AT) is one of the methods utilized most extensively. The map of an Arnold transformation was discovered in the 1960s by Vladimir Arnold using a cat image (Arnold and Avez 1968); the map is described in Eq. 1:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod 1 \tag{1}$$

where x and $y \in \{0, 1\}$. The formula illustrated above is defined for a unit square though which the existing matrix can be extended upon image pixels, i.e., if $x, y \in \{0, 1, 2, 3, \ldots., N\}$. With the increase in image pixels, there will also be an increase of elements in the matrix, and Eq. 1 can be rewritten as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{2}$$

An Arnold map (AM) utilizes linear algebra concepts on the positioning of pixels to change their values (Ye and Wong 2012). An AM can shuffle image pixels of any size and is generalizable. The generalized Arnold map (AMg) is expressed in matrix notation, as demonstrated by Eq. 3:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{3}$$

In this equation, a and b are the two control parameters that aid in changing the position of pixels x and y, making new coordinates of pixels x' and y' in the shuffled image. The pixels of original image (x, y) will then transform into shuffled pixels of (x', y').

On the other hand, the distinctive exponents of the Lyapunov exponent are calculated as shown by Eq. 4:

$$\lambda = 1 + \frac{ab + \sqrt{a^2b^2 + ab}}{2} > 1 \tag{4}$$

This map will behave chaotically if the Lyapunov exponent (LE) is greater than 1 (Ye 2011). This implies that if the a and b are each greater than 0, i.e., $(a > 0)$ and $(b > 0)$, then the system will be in a chaotic state.

The Arnold map generalized (AMg) equation is the discrete system that works on two effects, namely stretching and folding. These effects can be attained using the phase space system, which helps in creating confusing image encryption schemes. However, to obtain the randomly confused image, the confusion process is repeated several times. As a result, utilizing AMg as part of an image encryption scheme will take a long time. Furthermore, a digital image's finite gray levels may cause the original image to reemerge after several rounds of confusion (Wang et al. 2010).

2.2 Logistic may system

A logistic and may map (LOMAS) is a discrete time 1D chaotic system (Nkandeu and Tiedeu 2019) that can be achieved using Eq. 5:

$$y_{m+1} = (y_m e^{\wedge}((r'+9)(1-y_m)) - (r'+5)y_m(1 - y_m)) \bmod 1 \tag{5}$$

where $y_m \in [\,0\ 1]$ and $r' \in [0, 5]$. This modified system will behave with chaotic randomness.

3 DNA system

This section will discuss gene expression, DNA basics—i.e., the four nucleotides—and their application in image encryption.

3.1 DNA and gene expression

Gene expression is the continuous process by which the genome receives and decrypts information that the living organism can utilize and process using a DNA code (Tefferi 2006). The fundamental dogma of living organisms is responsible for gene expression. A DNA molecule is fed into the central dogma process, which is then synthesized into a polypeptide chain that possesses many amino acids bonded together. Molecular biology has also demonstrated that proteins are retrieved using DNA (Hollenbach 2020). Transcription and translation are the two critical stages of the central dogma process (Cooper 1981). Transcription turns DNA into RNA, while polypeptide chains can be obtained by converting RNA through translation. The core dogma process is depicted in Fig. 1.

3.2 DNA composition

DNA is composed of four nucleic acid bases. The human genome is enormously long and sophisticated, is comprised of approximately 3.2 billion base-paired nucleotides. These are the four most essential nucleotide bases (Watson and Crick 1953), which are adenine (A), cytosine (C), thymine (T), and guanine (G). These four nucleic acids are complementary pairs, i.e., like the binary '0' and '1,' they complement to each other. When seeking pairwise combinations, we can easily find that '00,' '01,' '10,' and '11' are complementary binary pairs. Thus, it is easy to encode binary numbers of '00,' '01,' '10,' '11' using four bases, i.e., 'A,' 'C,' 'G,' 'T.' Using 4! = 24, we can also get the maximum possible number of schemes. Eight out of 24 schemes have satisfied the complementary base pair principle shown in Table 1 (Watson and Crick 1993). DNA sequences have better encryption properties and meet all

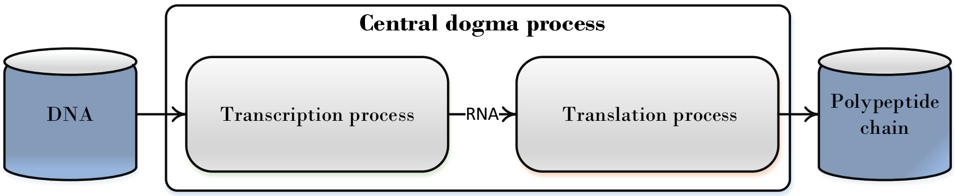

Fig. 1 The process of central dogma

Table 1 The relationship of four nucleotides with $P_{(i,j)}$

$P_{(i,j)}$	0	1	2	211	248	255
DNA code	[A|A|A|A]	[A|A|A|C]	[A|A|A|G]	[T|C|A|T]	–	–

Table 2 DNA coding rules (4! = 24)

R1: (0, 1, 2, 3)	R7: (1, 0, 2, 3)	R13: (2, 0, 1, 3)	R19: (3, 0, 1, 2)
R2: (0, 1, 3, 2)	R8: (1, 0, 3, 2)	R14: (2, 0, 3, 1)	R20: (3, 0, 2, 1)
R3: (0, 2, 1, 3)	R9: (1, 2, 0, 3)	R15: (2, 1, 0, 3)	R21: (3, 1, 0, 2)
R4: (0, 2, 3, 1)	R10: (1, 2, 3, 0)	R16: (2, 1, 3, 0)	R22: (3, 1, 2, 0)
R5: (0, 3, 1, 2)	R11: (1, 3, 0, 2)	R17: (2, 3, 0, 1)	R23: (3, 2, 0, 1)
R6: (0, 3, 2, 1)	R12: (1, 3, 2, 0)	R18: (2, 3, 1, 0)	R24: (3, 2, 1, 0)

$^{*}R = rule$

tests for constructed S-boxes, which in turn means these qualify for real-time communication.

Consider a colored plain image P having a size of $L \times W \times 3$ pixels. P is divided into three layers, i.e., red (R), green (G), and blue (B) layers, respectively. The image pixel $P_{(i,j)}$ depicts the position of each pixel where $i = 1, 2, 3, \ldots.$ L and $j = 1, 2, 3, \ldots. W$ each lie in between 0 and 255 for an eight-bit system. Thus, the DNA cryptography-based encoding scheme can be expressed as (Zhang 2018):

$$P_{(i,j)} = b_3.4^3 + b_2.4^2 + b_1.4^1 + b_0.4^0 \tag{6}$$

A, C, G, and T are represented as b_i, where i $= 0, 1, 2, 3$. As a result, each pixel may be assigned to the tetrads. The appropriate relationship of A, C, G, and T with $P_{(i,j)}$ is depicted in Table 1.

Zhang (2018) established 24 similar connections between 0, 1, 2, 3 and A, T, C, G, which they explained as 24 types of principles based on DNA computing and coding. The result is shown in Table 2. In these combinations we utilized A, T, C, G with the available rule number 6: (0, 3, 2, 1) in our proposed scheme. This combination can be represented by $A0, T3, C2, G1$. Zhang (2018) also presented various kinds of combination and operations, such as DNA join operations and compliment operations, out of 24 kinds of DNA coding rules, as shown in Table 3. Different methods, such as commutative and associative methods, can be applied. The exclusive OR operation for DNA rules is depicted in Tables 4 and 5.

Let us assume that there are three nucleic acids signified by x, y, and z. The DNA join operation between x and y is represented by x, y as shown in Eq. 7–9:

$$\langle x, y \rangle = \langle y, x \rangle \tag{7}$$

$$\langle x, y, z \rangle = \langle \langle x, y \rangle, z \rangle = \langle x, \langle y, z \rangle \tag{8}$$

$$\begin{bmatrix} A \\ T \\ C \\ G \end{bmatrix} = \begin{bmatrix} A & T & C & G \\ T & A & G & C \\ C & G & A & T \\ G & C & T & A \end{bmatrix} \triangle OP1 \tag{9}$$

3.3 Transcription process

In molecular biology, transcription is the process of generating RNA from DNA molecules (Hardy et al. 2004). In cryptography, the input sequence of DNA is changed to the output sequence of RNA using DNA sequencing. Using a version of the confusion process, the four nucleotide bases are substituted with the corresponding Watson–Crick (w–c) complements through transcription. For instance, the T is replaced with the U because the RNA strand lacks thymine (T), which is substituted by uracil (U). Likewise, T is swapped with A, A is changed with U, G is replaced with C, and C is replaced with G in the Watson–Crick complementary pairing scheme (Watson and Crick 1953, 1993).

3.4 DNA operation

3.4.1 DNA coding

The line vector based on a sequence of DNA (DSLV) can be achieved through the process of converting pixels' line vector utilizing 24 kinds of DNA encoding rules. This process can be expressed as in Eq. 10:

$$Rule = \mathrm{mod}\ (floor(mean(sub\_key), 24) \tag{10}$$

3.4.2 DNA chosen operations

The DNA system can utilize different operations as a means of generating sequences. The system can also be

Table 3 The 16 possible join operations

$\begin{bmatrix} A & T & C & G \\ T & A & G & C \\ C & G & A & T \\ G & C & T & A \end{bmatrix} \triangle OP1$	$\begin{bmatrix} A & T & C & G \\ T & A & G & C \\ C & G & T & A \\ G & C & A & T \end{bmatrix} \triangle OP2$	$\begin{bmatrix} A & T & C & G \\ T & C & G & A \\ C & G & A & T \\ G & A & T & C \end{bmatrix} \triangle OP3$	$\begin{bmatrix} A & T & C & G \\ T & G & A & C \\ C & A & G & T \\ G & C & T & A \end{bmatrix} \triangle OP4$
$\begin{bmatrix} T & A & G & C \\ A & T & C & G \\ G & C & A & T \\ C & G & T & A \end{bmatrix} \triangle OP5$	$\begin{bmatrix} T & A & G & C \\ A & T & C & G \\ G & C & T & A \\ C & G & A & T \end{bmatrix} \triangle OP6$	$\begin{bmatrix} T & C & G & A \\ C & G & A & T \\ G & A & T & C \\ A & T & C & G \end{bmatrix} \triangle OP7$	$\begin{bmatrix} T & G & A & C \\ G & C & T & A \\ A & T & C & G \\ C & A & G & T \end{bmatrix} \triangle OP8$
$\begin{bmatrix} C & A & G & T \\ A & T & C & G \\ G & C & T & A \\ T & G & A & C \end{bmatrix} \triangle OP9$	$\begin{bmatrix} C & G & A & T \\ G & A & T & C \\ A & T & C & G \\ T & C & G & A \end{bmatrix} \triangle OP10$	$\begin{bmatrix} C & G & A & T \\ G & C & T & A \\ A & T & C & G \\ T & A & G & C \end{bmatrix} \triangle OP11$	$\begin{bmatrix} C & G & T & A \\ G & C & A & T \\ T & A & G & C \\ A & T & C & G \end{bmatrix} \triangle OP12$
$\begin{bmatrix} G & A & T & C \\ A & T & C & G \\ T & C & G & A \\ C & G & A & T \end{bmatrix} \triangle OP13$	$\begin{bmatrix} G & C & A & T \\ C & A & G & T \\ T & G & A & C \\ A & T & C & G \end{bmatrix} \triangle OP14$	$\begin{bmatrix} G & C & T & A \\ C & A & G & T \\ T & G & A & C \\ A & T & C & G \end{bmatrix} \triangle OP15$	$\begin{bmatrix} G & C & T & A \\ C & G & A & T \\ T & A & G & C \\ A & T & C & G \end{bmatrix} \triangle OP16$

designed using various operations, such as XOR, ADD, and SUB. The resulting DNA system can be expressed as in Eq. 11:

$$OP = \bmod\ (floor(mean(Sub\_key), 3) \tag{11}$$

3.4.3 Number of rounds

The selected operation will calculate several rounds (NOR) using Eq. 12:

$$NOR = floor(\log(vectsize)\log(2)), \tag{12}$$

wherein $<<vectsize>>$ depicts the size of the DNA vector sequence.

3.4.4 DNA joined operation

In a DNA joined operation, the permutation sequence is transformed to a permuted sequence generated by DNA with the corresponding information DNA sequence using one of the 16 possible DNA join operations. This process can be expressed as in Eq. 12:

$$Rule = \bmod\ (floor(avg(sub\_key), 17) \tag{13}$$

4 Anticipated algorithm for the construction of S-box

1. Let T be a plain text image with $j \times k$ representing the entire dimension of a plain image where j and k are the image's rows and columns, respectively. The original image (T) having an initial size 512 × 512 × 3 pixels is divided into three channels (e.g., red = R, green = G, and blue = B) each containing 512 × 512 pixels. The three divided channels are then saved to U, where U = T(T_R, T_G, and T_B).
2. A two-dimensional Arnold transformation is iterated to generate random sequences X and then resized to an exact fit for the pixels of each channel T(T_R, T_G, and T_B) in U. The resized sequences are then saved to $X2$. Moreover, the sequences $X2$ are tested for several rounds, up to 256 in number $(R_1, R_2, R_3, \ldots R_{256})$, to achieve a shuffle matrix $(M_1, M_2, M_3, \ldots M_{256})$.
3. In this step, the divided channels (T_R, T_G, T_B) containing 262,144 pixels each are XORed with the resized sequences ($X2$) for three rounds $(R_1, R_2, andR_3)$ of the shuffled matrix $(M_1, M_2, andM_3)$.
4. In this step, random permutation (R_P) is initiated in order to generate maximum random sequences, which are then further treated with the three rounds $(R_1, R_2, andR_3)$ of the shuffled matrix $(M_1, M_2, andM_3)$ to achieve V, which is constituted of $(R_P \oplus M_1, M_2, andM_3)$.
5. An alphabetical DNA amino sequence (W) is generated and then converted into four nucleotides, the adenine (A), thymine (T), cytosine (C), and guanine (G), where $W = (A, T, C, G)$. The values are encoded as $W_{encoded}(A = 00,\ C = 01,\ G = 10$, and $T = 11)$, as shown in Fig. 2.
6. The encoded values for the DNA-based $W_{encoded} = (00, 01, 10, 11)$ is furthermore quantized ($Q =$), with $Q = 0.5$, 1.5, 1, and 0. The output is then stored in Y.
7. A modified logistic–may map $LM_{RS} = y_{m+1} = (y_m e \wedge ((r' + 9)(1 - y_m)) - (r' + 5) \qquad y_m(1 - y_m))$mod 1 is then generated, including a definition

Table 4 Exclusive OR operation for DNA rules

Rules (1,2,3,4,5,6)					Rules (7,8,13,14,19,20)				
Exclusive OR (XOR)	A	T	C	G	Exclusive OR (XOR)	A	T	C	G
A	A	T	C	G	A	T	A	G	C
T	T	A	G	C	T	A	T	C	G
C	C	G	A	T	C	G	C	T	A
G	G	C	T	A	G	C	G	A	T
Rules (9,11,15,17,21,23)					Rules (10,12,16,18,22,24)				
Exclusive OR (XOR)	A	T	C	G	Exclusive OR (XOR)	A	T	C	G
A	C	G	A	T	A	G	C	T	A
T	G	C	T	A	T	C	G	A	T
C	A	T	C	G	C	T	A	G	C
G	T	A	G	C	G	A	T	C	G

Table 5 XOR operation for DNA genetic sequence (Rules (1,2,3,4,5,6))

XOR	A	C	G	T
A	A	T	C	G
C	T	A	G	C
G	C	G	A	T
T	G	C	T	A

of its initial conditions. This map is iterated in order to generate equal numbers of random sequences (*RS*) to that of the original pixels of an image (*U*). Moreover, $LM_{(RS)}$ is reshaped to a matrix having 512 × 512 pixels equal to plain text image $T = j \times k$ and is stored to *RM*.

8. The *RM* takes the modulus and round functions and is converted into *unit8* integers. Then the result of *RM* is stored in $Y1$.
9. Steps 7 and 8 are repeated for the stored values of Y to get DNA random sequencing by applying the round and modulus functions, respectively, to achieve an output ($Z0$).

Table 6 The obtained S-box

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

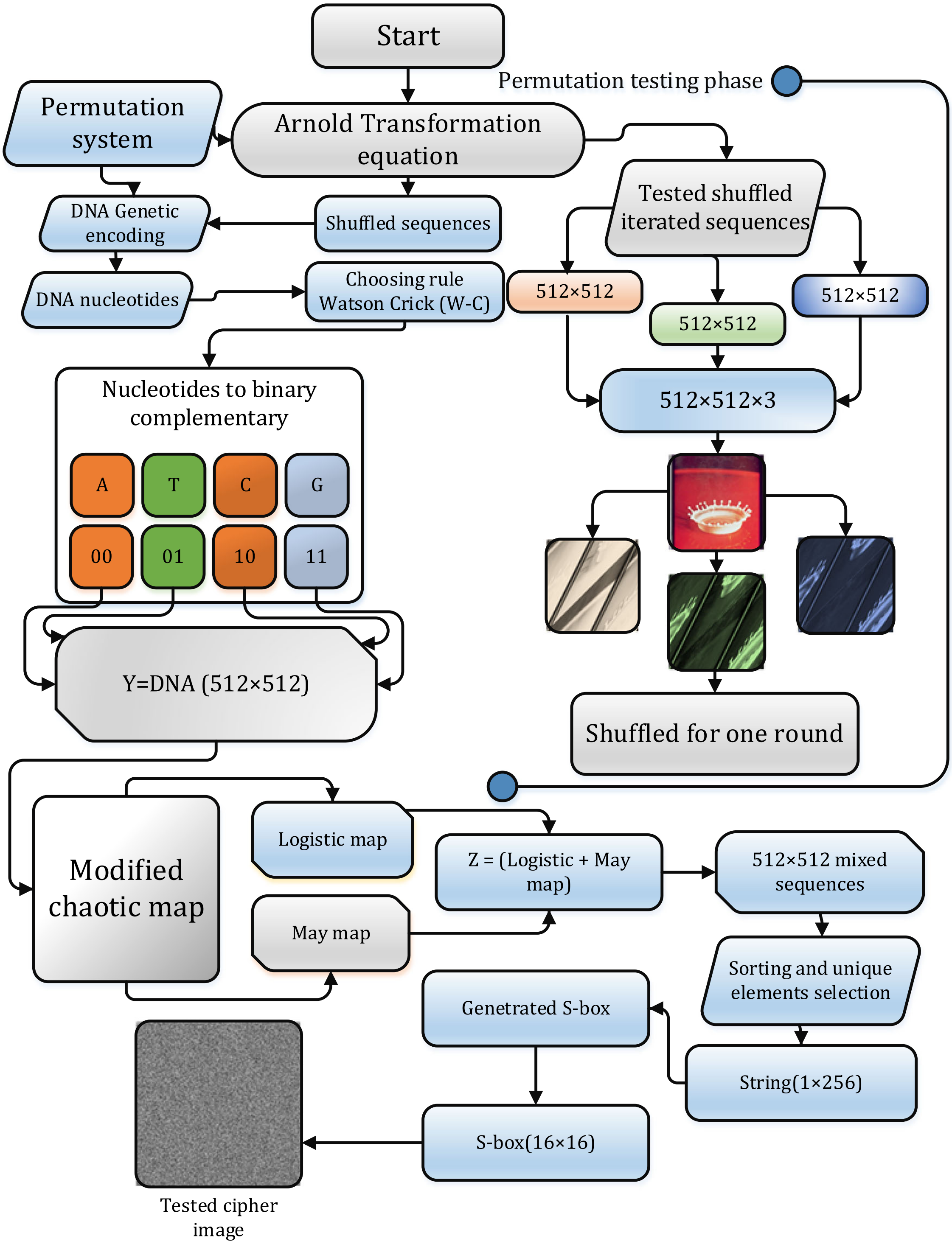

Fig. 2 Block diagram demonstrating the proposed S-box based on DNA and chaos sequencing

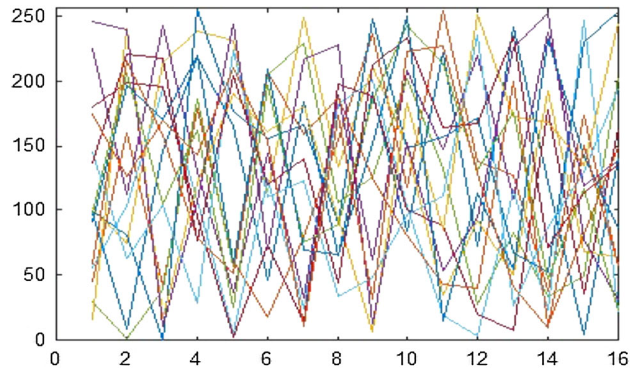

Fig. 3 Plot of the constructed S-box

Table 7 NIST 800-22 test results for the obtained S-box

NIST 800-22 tests	*p* values	Result
Frequency test		Success
Block frequency test		Success
Cumulative sums tests		Success
Runs test		Success
Longest run of one's test		Success
Rank test		Success
Discrete Fourier transform test		Success
Approximate entropy test		Success
Serial test		Success
Linear complexity test		Success
Random excursions tests		N/A
Random excursion variant test		N/A
Universal statistical test		N/A

10. The XOR operation is applied between the stored values of $Y1$ and $Z0$ to obtain $Z1$. Or, in other words, $Z1 = Y1 \oplus Z0$.
11. By this step, we have obtained the S-box through the selection of unique elements of $Z1$, such as how $Z1$ became $S1(1 \times 256)$ and random sequencing array is sorted into either ascending or descending order.
12. Finally, the array $S2(1 \times 256)$ is reshaped to $S_{obtained} = S2(16 \times 16)$.

The output of this twelve-step process is shown in Table 6 (Matthews 1989), while the plot of the S-box thus constructed is shown in Fig. 3.

5 Randomness tests for the constructed S-box

5.1 NIST test

Table 7 depicts the completed S-box. To test and determine its level of randomness, the National Institute of Standards and Technology (NIST-800-22) statistical tests are used, including several closely interdependent tests and a close examination of any non-randomness that may exist following the proposed generated sequence. The outcomes of these statistical tests are then assessed according to the *p* value also established by the NIST-800-22, which holds that the resulting *p* value must be either equal to or more significant than the present value to signify success. Table 7 shows the results of several randomization tests (Pareschi et al. 2012).

5.2 Histogram uniformity analysis

This analysis is used to assess the pixel arrangement of each channel. The regularity of the pixels is determined by the randomization (i.e., random numbers) obtained using our proposed S-box. The pixels' non-uniformity intimates that the system is not secure, and the data is easily retrievable. As discussed earlier in Sect. 4, let T be a plain text image. The original image (T) of 512 × 512 × 3 pixels is divided into three channels, each containing 512 × 512 pixels. The constructed S-box is then applied on all three channels, as shown in Figs. 4–8.

Figure 4 depicts that the initial image is divided into three respective channels. The Arnold transformation method is utilized to obtain a shuffled image, as shown in Fig. 5. The intensity of shuffling depends upon the number of rounds applied. However, shuffling indicates that the permutation of pixels does not change the distribution of pixels in each channel, as shown in Fig. 6. The pixels show uniformity when the substitution method is applied, as shown in Figs. 7 and 8. A high level of randomness is achieved by the addition of a modified LOMAS and our DNA-based S-box.

6 S-box fundamental characteristics and experimentation process

The robustness and performance of our proposed S-box are also assessed using five different industry standard tests, including the nonlinearity test (N-L), strict avalanche criterion (SAC), bit independence criterion (BIC), differential approximation probability (DP), and linear approximation probability (LP). As the results below demonstrate, each

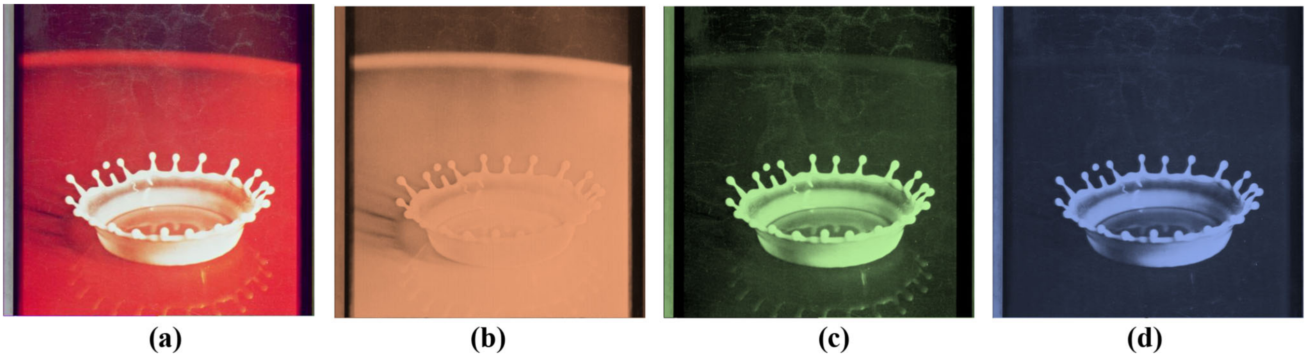

Fig. 4 **a** Plain image of splash (512 × 512 × 3); **b** red channel; **c** green channel; **d** blue channel

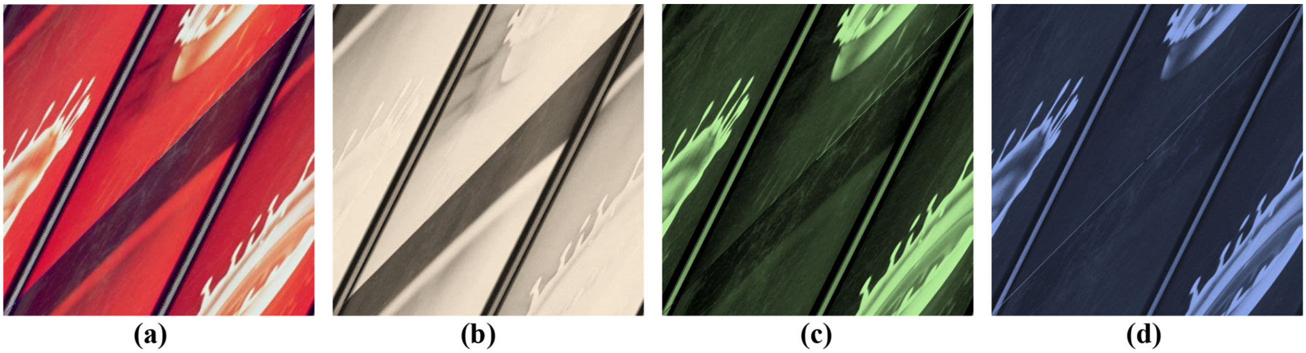

Fig. 5 **a** Shuffled plain image of splash (512 × 512 × 3); **a** red channel; **b** green channel; **c** blue channel

test yielded exceptional results, indicating that the S-box we have constructed competes well against existing S-boxes and even delivers a more remarkable ability to withstand linear assaults.

6.1 Nonlinearity test analysis

The strength of the encryption achieved for various data using the substitution process can be evaluated using the nonlinearity test. The original data is already distorted by substituting pixels of the original image with the constructed S-box. The discussed criterion can be illustrated by using a Boolean function $g(x)$, whose nonlinearity can be defined as:

$$N_g = 2^{k-1}(1 - 2^k \max_{\varphi \in GF(2^k)} |S_{(g)}(\varphi)|), \tag{14}$$

$$S_{(g)}(\varphi) = \sum_{\varphi \in GF(2^k)} (-1)^{x.\varphi \oplus g(x)}, \tag{15}$$

where $x.\varphi = x_1 \oplus \varphi_1 + x_2 \oplus \varphi_2 + \cdots + x_n \oplus \varphi_n$. The Boolean function $g(x)$ and nonlinearity N_g each have a direct relation to each other, so that if the value of N_g increases, then the value of $g(x)$ will also increase. As a result, the S-box's ability to resist any linear passwords will be robust. If the amount of nonlinearity introduced by an S-box is not sufficient to protect against linear attacks, then unauthorized users can understand the behavior of the Boolean function. Thus, the strength of selected bits in the Boolean function is the fundamental cause of changes in these characteristics.

In this test, we analyzed our proposed S-box's changing values by changing bits in the corresponding Boolean process. The results and comparison of different S-boxes are shown in Table 8.

6.2 Strict avalanche criterion (SAC) test analysis

The strict avalanche effect is a criterion in which the changes in bits are proportional to the number of bits in the encrypted message. In this way, the encrypted message will change dramatically if the plain text or key is altered. When complementing a single bit at the input, the strict avalanche effect (SAC) condition is satisfied, and all the output bits will change with half the probability. This change in the input causes an avalanche effect that spreads throughout the system.

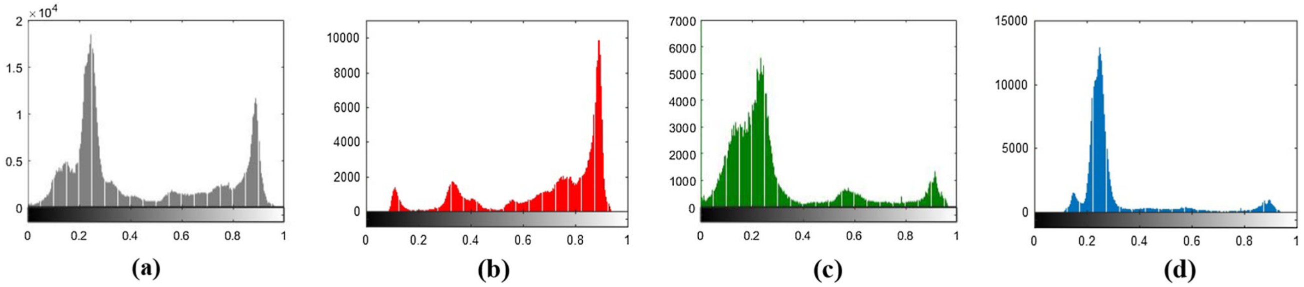

Fig. 6 **a** Plain and shuffled image histogram (512 × 512 × 3); **a** red channel histogram; **b** green channel histogram; **c** blue channel histogram

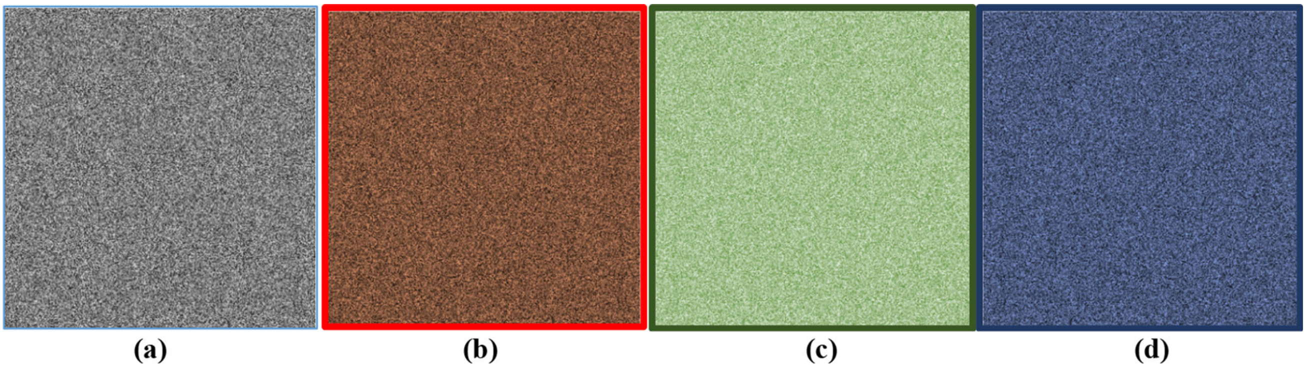

Fig. 7 **a** Encrypted image of splash (512 × 512 × 3); **b** red channel; **c** green channel; **d** blue channel

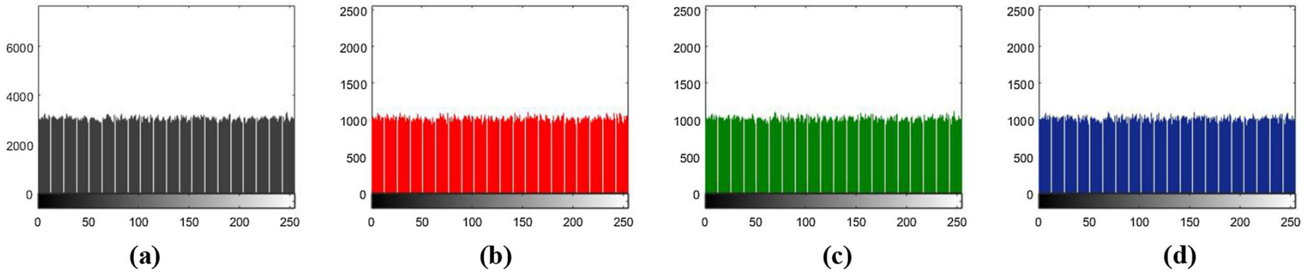

Fig. 8 **a** Encrypted image histogram (512 × 512 × 3); **b** red channel histogram; **c** green channel histogram; **d** blue channel histogram

By adjusting the input bits, we were able to examine the avalanche effect regarding our proposed S-box. Here it is essential to have a dependence matrix with all the values of dependence averaging 0.5. This dependence matrix can be evaluated using Eqs. 16–17:

$$S(g) = \frac{1}{k^2} \sum_{1<r\leq k} \sum_{1\leq\omega\leq k} \left| \frac{1}{2} - Q_{r,\omega}(g) \right|, \tag{16}$$

$$Q_{r,\omega}(g) = 2^{-k} \sum\nolimits_{x\in B^k} g_\omega(x) \oplus g_\omega(x \oplus e_r). \tag{17}$$

where $e^r = [\theta_{r,1}\theta_{r,2}...\theta_{r,k}]^T$ and $\theta_{r,\omega} = 0$ when $r \neq \omega$ and $\theta_{r,\omega} = 1$ when $r = \omega$.

Table 9 shows the dependence matrix for testing using SAC. The values range from highest to lowest, with the S-box we created showing a maximum resultant value of 0.5625 and a minimum value of 0. 4375. This means that our S-mean box's dependence matrix value is 0.5022, close to the SAC optimum value of 0.5. SAC values for different S-boxes are reported in Table 10. When compared to current S-boxes, our proposed S-box offers better performance and throughput.

Table 8 Nonlinearity test for S-boxes

S-box	Max	Min	Mean
Our S-box	106	98	103.75
Jakimoski et al. S-box (Jakimoski and Kocarev 2001)	108	100	103.25
Khan et al. S-box (Khan et al. 2012)	106	96	103
Tang et al. S-box (Tang et al. 2005)	109	103	104.88
Chen et al. S-box (Chen et al. 2007)	106	100	103
Ozkaynak et al. S-box (Özkaynak and Özer 2010)	106	100	103.25
Hussain et al. S-box (Hussain et al. 2012)	108	102	104.75
Khan et al. S-box (Khan et al. 2013)	108	98	103
Lambic et al. S-box (Lambić 2014)	112	108	109.25
Liu et al. S-box (Liu et al. 2015)	108	104	105.8
Belazi et al. S-box (Belazi and El-Latif 2017)	110	102	105.5
Belazi et al. S-box (Ullah et al. 2017)	108	102	106
Al Solami et al. S-box (Al Solami et al. 2018)	110	106	108.5

Table 9 Dependence matrix of constructed S-box

0.5547	0.5625	0.5000	0.5000	0.4844	0.5469	0.5547	0.5000
0.5547	0.5000	0.5078	0.5000	0.5000	0.5469	0.4453	0.4453
0.5547	0.5625	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
0.5000	0.4375	0.4922	0.5000	0.5000	0.5000	0.5547	0.4453
0.4453	0.5625	0.4922	0.5000	0.4844	0.5000	0.5000	0.4453
0.4453	0.4375	0.5000	0.5000	0.5156	0.5000	0.5000	0.5547
0.5000	0.5000	0.5078	0.5000	0.5000	0.5469	0.5000	0.5000
0.5000	0.4375	0.5000	0.5000	0.5156	0.5000	0.5000	0.5000

Table 10 Analysis of strict avalanche criterion (SAC) for different S-boxes

S-boxes	Max	Min	Mean
Our obtained S-box	0.5625	0.4375	0.5022
Jakimoski et al. S-box (Jakimoski and Kocarev 2001)	0.5938	0.3750	0.5059
Tang et al. S-box (Tang et al. 2005)	0.5703	0.3984	0.4966
Chen et al. S-box (Chen et al. 2007)	0.6094	0.4219	0.5000
Ozkaynak et al. S-box (Özkaynak and Özer 2010)	0.5938	0.4219	0.5049
Hussain et al. S-box (Hussain et al. 2012)	0.5938	0.3906	0.5056
Khan et al. S-box (Khan et al. 2013)	0.5938	0.4063	0.5012
Lambic et al. S-box (Lambić 2014)	0.5937	0.4375	0.5012
Liu et al. S-box (Liu et al. 2015)	0.5938	0.4219	0.4976
Belazi et al. S-box (Belazi and El-Latif 2017)	0.5925	0.4375	0.5000
Belazi et al. S-box (Belazi et al. 2017)	0.5313	0.4297	0.4956
Khan et al. S-box (Khan et al. 2012)	0.6250	0.3906	0.5039
Al Solami et al. S-box (Al Solami et al. 2018)	0.5937	0.4062	0.5017

6.3 Bit independence criterion (BIC) test analysis

The bit independence criterion (BIC) test is one of the essential criteria used to measure of S-boxes' strength. With this test, there must be reasonable independence of change in bit pattern at the output end from the input end. If this is the case, it becomes difficult for intruders to map the change criteria in input to output bits. If the S-box in question satisfies the BIC test properties, there must be an independent pairwise avalanche variable for the definite

Table 11 Bit independence criterion (BIC)–nonlinearity (NL) for constructed S-box

–	104	116	110	110	104	108	116
104	–	120	108	110	114	114	118
116	120	–	112	116	108	112	112
110	108	112	–	116	116	114	112
110	110	116	116	–	118	112	114
104	114	108	116	118	–	110	106
108	114	112	114	112	110	–	116
116	118	112	112	114	106	116	–

order of avalanche vectors. The avalanche criterion is based on changing input bits and analyzing the nature of output bits on the other side. This is mandatory for bit independence criterion (BIC) property, which should use $g_r \oplus g_w (r \neq w, 1 \leq r, w \leq n)$ to accomplish nonlinearity.

The results that we achieved are shown in Tables 11 and 12, respectively. The obtained values for BIC-SAC and BIC-nonlinearity for our S-box are 0.4960 and 112.35, indicating that our S-box fulfills the requirements of both bit independence criterion strict avalanche criterion (BIC-SAC) and bit independence criterion nonlinearity (BIC-NL) properties. Table 13 then depicts BIC properties for various existing S-boxes, demonstrating that the values our new S-box achieved for BIC-SAC and BIC-nonlinearity are higher than those of existing S-boxes.

6.4 The Equiprobable input/output *XOR* distribution

This test measures the variation of bits at the output in response to the variation of bits at the input. The differential approximation probability can be obtained when any change of ∂k at the input will immediately change ∂l at the output. However, it is also important to note that the likelihood of input XOR values and output XOR values must be the same. Ideally, the S-box should have differential uniformity, which can be obtained using differential probability (DP), as illustrated thus:

$$DP_{g(\partial k \to \partial l)} = \left[\frac{\#\{k \in \frac{X}{S(k)} \oplus S(k \oplus \partial k) = \partial l\}}{2^m} \right] \quad (18)$$

Table 12 Bit independence criterion (BIC)–strict avalanche criterion (SAC) for constructed S-box

0.5000	0.4955	0.4955	0.5000	0.4933	0.5045	0.4933	0.4911
0.4978	0.4911	0.4933	0.4866	0.4799	0.5022	0.4955	0.4955
0.5045	0.4922	0.4955	0.4877	0.4944	0.5011	0.4933	0.5000
0.5056	0.4944	0.4922	0.4877	0.4922	0.4877	0.4833	0.5033
0.4967	0.4967	0.5078	0.4944	0.5056	0.4989	0.4989	0.4967
0.4944	0.5000	0.4922	0.5067	0.4911	0.5000	0.5033	0.5033
0.4944	0.4967	0.4877	0.4967	0.4989	0.4922	0.4922	0.5011
0.5022	0.5000	0.5067	0.4866	0.4844	0.4978	0.4955	0.4955

Table 13 Comparison of Bit independence criterion (BIC)–nonlinearity (NL) with different S-boxes

S-box	BIC-SAC	BIC-Nonlinearity
Constructed S-box	0.4960	112.35
Jakimoski et al. S-box (Jakimoski and Kocarev 2001)	0.5031	104.29
Tang et al. S-box (Tang et al. 2005)	0.5044	102.96
Chen et al. S-box (Chen et al. 2007)	0.5024	103.14
Ozkaynak et al. S-box (Özkaynak and Özer 2010)	0.5010	103.71
Hussain et al. S-box (Hussain et al. 2012)	0.5022	104.07
Khan et al. S-box (Khan et al. 2013)	0.4989	104.07
Lambic et al. S-box (Lambić 2014)	–	108.21
Liu et al. S-box (Liu et al. 2015)	0.5032	104.5
Belazi et al. S-box (Belazi and El-Latif 2017)	0.4970	103.78
Belazi et al. S-box (Belazi et al. 2017)	0.4996	103.8
Ullah et al. S-box (Ullah et al. 2017)	0.5050	103
Khan et al. S-box (Khan et al. 2012)	0.5010	100.36
Al Solami et al. S-box (Al Solami et al. 2018)	0.5006	104

Table 14 Differential approximation for constructed S-box

6	6	6	6	6	10	6	6	6	6	6	6	8	8	8	6
10	6	10	6	8	6	8	6	6	6	6	8	6	8	6	8
8	6	6	8	10	8	8	6	6	6	8	6	8	8	6	8
6	6	8	6	8	6	6	6	8	8	6	10	6	6	6	8
8	8	8	6	8	8	10	6	6	6	10	6	8	4	8	6
8	6	6	6	6	8	8	8	8	6	8	8	6	6	6	6
6	6	6	8	6	6	6	6	6	6	8	6	6	6	8	6
6	8	6	8	6	6	8	6	8	8	6	10	8	6	6	6
6	8	8	8	8	6	6	6	6	6	6	8	6	8	6	6
6	10	8	6	6	6	6	6	6	8	8	10	6	6	6	6
6	10	8	8	6	6	8	8	8	6	6	8	6	6	8	6
6	8	6	6	6	10	8	8	8	6	6	6	8	8	6	8
10	6	6	10	6	6	8	6	6	6	6	6	6	6	6	6
6	6	6	6	6	6	4	10	8	8	8	6	4	8	6	6
8	6	6	6	6	8	6	6	8	10	6	6	6	6	6	8
6	6	10	4	8	6	6	8	6	6	8	6	8	8	6	-

Table 15 Comparison of linear approximation probability with different S-boxes

S-boxes	MaxLp
Constructed S-box	0.1560
AES S-box	0.062
APA S-box	0.062
Skipjack S-box	0.109
Xyi S-box	0.156
Jakimoski et al. S-box (Jakimoski and Kocarev 2001)	0.1250
Tang et al. S-box (Tang et al. 2005)	0.1328
Chen S-box (Chen et al. 2007)	0.1328
Ozkaynak et al. S-box (Özkaynak and Özer 2010)	0.1328
Hussain et al. S-box (Hussain et al. 2012)	0.1250
Khan et al. S-box (Khan et al. 2013)	0.1484
Belazi et al. S-box (Belazi and El-Latif 2017)	0.1250
Belazi et al. S-box (Belazi et al. 2017)	0.1562
Ullah et al. S-box (Ullah et al. 2017)	0.1250
Khan et al. S-box (Khan et al. 2012)	0.1484

where X is all the possible input values and 2^m counts several elements in constructed S-box. The smaller the value of DP_g the more robustly that S-box has been designed and the more vital its abilities to defend against differential attacks.

The differential approximation that we achieved is shown in Table 14. The maximum value of '10' indicates that a particular S-box has a good ability to resist differential attacks.

6.5 Linear approximation probability

Linear approximation probability (LP) can be defined as the highest unbalance value, which can be determined using the simple equation:

$$LP = \max_{\gamma_1, \gamma_2 \neq 0} \left(\frac{\#\{z \in Z | z.\gamma_1 = g(z).\gamma_2}{2^n} - \frac{1}{2} \right), \quad (19)$$

Here, the denotation γ_1 signifies the input mask and the γ_2 represents the output mask. Z is the representation of all possible input values while 2^n is the tally S-box elements. The parity of chosen mask γ_1 for the input bits is equal to parity of chosen mask γ_2 for the output bits. The LP value must be smaller in order to be stronger against any linear password attack.

The value of LP that we obtained using our new S-box is shown in Table 15, where it is also compared to existing S-boxes' LP values. As that table demonstrated, our S-box achieves a lower minimum LP value as compared to existing S-boxes' results, making ours the best performance.

7 Conclusion, discussion, and future prospects

In this article, we proposed a new S-box based on confusion and diffusion to protect sensitive visual information within images. We utilized Arnold cat map, DNA, and LOMAS sequences to achieve confusion and diffusion. Shuffling of the pixels of the plaintext image is achieved using Arnold map at specific iterations to achieve the permutation of pixels. The sequence-based on diffusion is achieved using bitwise XORed operation with DNA and LOMAS random sequences. Moreover, highly random 512 × 512 sequences are filtered to unique 256 × 1 elements, after which it is sorted into ascending order. Finally, the sorted output is reshaped to $S(16 \times 16)$ S-box. The newly constructed S-box utilizing various tests to assess and validate its randomness, all of which have been verified by NIST-800–22. The performance analysis we obtained from these different tests showed better results for our constructed S-box, which validate its cryptographic performance and demonstrate that it possesses a better ability to resist any attacks than most existing options. Moreover, we compared the results achieved by our new S-box with those from existing S-boxes for each of these different tests and performed various security analysis. The comparison showed that our constructed S-box has dominant cryptographic features. In future, our aim is to extend this research for audio and video encryption. We will investigate the proposed encryption technique for both audio and video data. The proposed scheme will also be tested on the work presented in Driss et al. (2020a), Masood et al. (2020c) and Driss et al. (2020b).

Acknowledgements The authors extend their gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under grant number R.G.P. 1/77/42.

Authors' contributions Investigation, software and simulation, writing original draft, writing review, and editing were performed by Fawad Masood and Junaid Masood. Conceptualization, methodology by Fawad Masood, and Jawad Ahmad. Supervision by Jawad Ahmad, Lejun Zhang and Wadii Boulilla. Writing review and editing were performed by Sadaqat Ur Rehman, Wadii Boulilla, Fadia Ali Khan and Sajjad Shaukat Jamal. All authors read and approved the final manuscript.

Funding One of the author Sajjad Shaukat Jamal extend his gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under Grant Number R. G. P. 1/77/42.

Declarations

Conflict of interest The authors wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Human and animals rights This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Not applicable.

References

Adleman LM (1994) Molecular computation of solutions to combinatorial problems. Science 266(5187):1021–1024

Adleman LM (1998) Computing with DNA. Sci Am 54–61

Ahmad J, Masood F, Shah SA, Jamal SS, Hussain I (2020) A novel secure occupancy monitoring scheme based on multi-chaos mapping. Symmetry 12(3):350

Al Solami E, Ahmad M, Volos C, Doja M, Beg M (2018) A new hyperchaotic system-based design for efficient bijective substitution-boxes. Entropy 20(7):525

Al-Najjar HM (2012) Digital image encryption algorithm based on multi-dimensional chaotic system and pixels location. Int J Comput Theory Eng 4(3):357

Arnold VI, Avez A (1968) Ergodic problems of classical mechanics, vol 9. Benjamin

Azam NA, Hayat U, Ayub M (2021) A substitution box generator, its analysis, and applications in image encryption. Signal Process 187:108144

Banthia AK, Tiwari N (2013) Image encryption using pseudo random number generators. Int J Comput Appl 67(20).

Belazi A, El-Latif AAA (2017) A simple yet efficient S-box method based on chaotic sine map. Optik Int J Light Electron Opt 130:1438–1444

Belazi A, Khan M, El-Latif AAA, Belghith S (2017) Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption. Nonlinear Dyn 87(1):337–361

Butt KK, Li G, Masood F, Khan S (2020) A digital image confidentiality scheme based on pseudo-quantum chaos and lucas sequence. Entropy 22(11):1276

COOper, S. (1981) The central dogma of cell biology. Cell Biol Int Rep 5(6):539–549

Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213

Chen G, Chen Y, Liao X (2007) An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps. Chaos, Solitons Fractals 31(3):571–579

Chen Y, Liao X (2005) Cryptanalysis on a modified Baptista-type cryptosystem with chaotic masking algorithm. Phys Lett A 342(5–6):389–396

Chen J, Zhu ZL, Zhang LB, Zhang Y, Yang BQ (2018) Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption. Signal Process 142:340–353

Chuang CH, Yen ZY, Lin GS, Hong ZW (2011) A virtual optical encryption software system for image security. J Converg Inf Technol 6(2).

Clelland CT, Risca V, Bancroft C (1999) Hiding messages in DNA microdots. Nature 399(6736):533–534

Daemen J, Rijmen V (1998) The block cipher Rijndael. In: International conference on smart card research and advanced applications. Springer, Berlin, pp 277–284

Driss K, Boulila W, Leborgne A, Gançarski P (2020) Mining frequent approximate patterns in large networks. Int J Imag Syst Technol

Driss K, Boulila W, Batool A, Ahmad J (2020) A novel approach for classifying diabetes' patients based on imputation and machine learning. In: 2020 international conference on UK-China Emerging Technologies (UCET). IEEE, pp 1–4

El Hanouti I, El Fadili H, Souhail W, Masood F (2020) A lightweight pseudo-random number generator based on a robust chaotic map. In: 2020 4th international conference on intelligent computing in data sciences (ICDS). IEEE, pp 1–6

Enayatifar R, Abdullah AH, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Opt Lasers Eng 56:83–93

El Hanouti I, El Fadili H (2021) Security analysis of an audio data encryption scheme based on key chaining and DNA encoding. Multimed Tools Appl 80(8):12077–12099

El Hanouti I, El Fadili H, Zenkouar K (2021a) Breaking an image encryption scheme based on Arnold map and Lucas series. Multimed Tools Appl 80(4):4975–4997

El Hanouti I, El Fadili H, Zenkouar K (2021b) Cryptanalysis of an embedded systems' image encryption. Multimedia Tools Appl 80(9):13801–13820

Hardy CD, Crisona NJ, Stone MD, Cozzarelli NR (2004) Disentangling DNA during replication: a tale of two strands. Philos Trans R Soc Lond Ser B Biol Sci 359(1441):39–47

Hollenbach AD (2020) Molecular genetics—the basics of gene expression. In: Clinical precision medicine. Academic Press, pp 11–26

Hussain I, Shah T, Gondal MA (2012) A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm. Nonlinear Dyn 70(3):1791–1794

Jakimoski G, Kocarev L (2001) Chaos and cryptography: block encryption ciphers based on chaotic maps. IEEE Trans Circuits Syst i: Fundam Theory Appl 48(2):163–169

Jiao S, Goutte R (2008) Code for encryption hiding data into genomic DNA of living organisms. In: 2008 9th international conference on signal processing. IEEE, pp. 2166–2169

Khan M, Shah T, Mahmood H, Gondal MA (2013) An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn 71(3):489–492

Khan M, Shah T, Mahmood H, Gondal MA, Hussain I (2012) A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems. Nonlinear Dyn 70(3):2303–2311

Lambić D (2014) A novel method of S-box design based on chaotic map and composition method. Chaos, Solitons Fractals 58:16–21

Li Y, Wang C, Chen H (2017) A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng 90:238–246

Liu X, Tong X, Wang Z, Zhang M (2021) Efficient high nonlinearity S-box generating algorithm based on third-order nonlinear digital filter. Chaos Solitons Fractals 150:111109

Liu G, Yang W, Liu W, Dai Y (2015) Designing S-boxes based on 3-D four-wing autonomous chaotic system. Nonlinear Dyn 82(4):1867–1877

Masood F, Ahmad J, Shah SA, Jamal SS, Hussain I (2020a) A novel hybrid secure image encryption based on julia set of fractals and 3D Lorenz chaotic map. Entropy 22(3):274

Masood F, Boulila W, Ahmad J, Sankar S, Rubaiee S, Buchanan WJ (2020b) A novel privacy approach of digital aerial images based on mersenne twister method with DNA genetic encoding and chaos. Remote Sensing 12(11):1893

Masood F, Driss M, Boulila W, Ahmad J, Rehman SU, Jan SU, Qayyum A, Buchanan WJ (2021) A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations. Wirel Person Commun 1–28

Masood J, Shahzad M, Khan ZA, Akre V, Rajan A, Ahmed S, Masood F (2020) Effective classification algorithms and feature selection for bio-medical data using IoT. In: 2020 7th international conference on information technology trends (ITT). IEEE, pp 42–47

Matthews R (1989) On the derivation of a "chaotic" encryption algorithm. Cryptologia 13(1):29–42

Mondal B, Mandal T (2017) A light weight secure image encryption scheme based on chaos & DNA computing. J King Saud Univ Comput Inf Sci 29(4):499–504

Munir N, Khan M, Wei Z, Akgul A, Amin M, Hussain I (2020) Circuit implementation of 3D chaotic self-exciting single-disk homopolar dynamo and its application in digital image confidentiality. Wirel Netw 1–18.

Munir N, Khan M, Hazzazi MM, AIjaedi A, Ismail AH, Alharbi AR, Hussain I (2021) Cryptanalysis of internet of health things encryption scheme based on chaotic maps. IEEE Access.

Munir N, Khan M, Jamal SS, Hazzazi MM, Hussain I (2021) Cryptanalysis of hybrid secure image encryption based on Julia set fractals and three-dimensional Lorenz chaotic map. Math Comput Simul

Naskar PK, Chaudhuri A (2016) Secured secret sharing technique based on chaotic map and DNA encoding with application on secret image. Imaging Sci J 64(8):460–470

Nkandeu YPK, Tiedeu A (2019) An image encryption algorithm based on substitution technique and chaos mixing. Multimed Tools Appl 78(8):10013–10034

Özkaynak F, Özer AB (2010) A method for designing strong S-Boxes based on chaotic Lorenz system. Phys Lett A 374(36):3733–3738

Pareschi F, Rovatti R, Setti G (2012) On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution. IEEE Trans Inf Forensics Secur 7(2):491–505

Qayyum A, Ahmad J, Boulila W, Rubaiee S, Masood F, Khan F, Buchanan WJ (2020) Chaos-based confusion and diffusion of image pixels using dynamic substitution. IEEE Access 8:140876–140895

Rivest RL (1990) Cryptography. In: Algorithms and complexity, pp 717–755

Sani RH, Behnia S, Akhshani A (2021) Creation of S-box based on a hierarchy of Julia sets: image encryption approach. Multidimens Syst Signal Process 1–24

Stallings W (2006) Cryptography and network security, 4/E. Pearson Education India, Delhi

Tang G, Liao X, Chen Y (2005) A novel method for designing S-boxes based on chaotic maps. Chaos Solitons Fractals 23(2):413–419

Tefferi A (2006) Genomics basics: DNA structure, gene expression, cloning, genetic mapping, and molecular tests. In: Seminars in cardiothoracic and vascular anesthesia, vol 10(4). Sage, Thousand Oaks, pp 282–290

Ullah A, Jamal SS, Shah T (2017) A novel construction of substitution box using a combination of chaotic maps with improved chaotic range. Nonlinear Dyn 88:2757–2769

Wang XY, Yang L, Liu R, Kadir A (2010) A chaotic image encryption algorithm based on perceptron model. Nonlinear Dyn 62(3):615–621

Watson JD, Crick FH (1993) Genetical implications of the structure of deoxyribonucleic acid. JAMA 269(15):1967–1969

Watson JD, Crick FH (1953) The structure of DNA. In: Cold Spring Harbor symposia on quantitative biology, vol 18. Cold Spring Harbor Laboratory Press, pp 123–131

Wheeler DD, Matthews RA (1991) Supercomputer investigations of a chaotic encryption algorithm. Cryptologia 15(2):140–152

Ye R (2011) A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. Opt Commun 284(22):5290–5298

Ye G, Wong K-W (2012) An efficient chaotic image encryption algorithm based on a generalized Arnold map. Nonlinear Dyn 69(4):2079–2087. https://doi.org/10.1007/s11071-012-0409-z

Zahid AH, Ahmad M, Alkhayyat A, Hassan MT, Manzoor A, Farhan AK (2021) Efficient dynamic S-box generation using linear trigonometric transformation for security applications. IEEE Access

Zhang Y (2018) The image encryption algorithm based on chaos and DNA computing. Multimed Tools Appl 77(16):21589–21615

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Fawad Masood[1,9] · **Junaid Masood**[2] · **Lejun Zhang**[1] · **Sajjad Shaukat Jamal**[3] · **Wadii Boulila**[4,5] · **Sadaqat Ur Rehman**[6] · **Fadia Ali Khan**[7,8] · **Jawad Ahmad**[9]

✉ Jawad Ahmad
J.Ahmad@napier.ac.uk

[1] College of Information Engineering, Yangzhou University, Yangzhou 225009, China

[2] Department of Computer Science, IQRA National University Peshawar, Peshawar, Pakistan

[3] Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia

[4] Robotics and Internet of Things Lab, Prince Sultan University, Riyadh, Saudi Arabia

[5] RIADI Laboratory, University of Manouba, 2010 Manouba, Tunisia

[6] Department of Computer Science, Namal Institute, Mianwali, Pakistan

[7] Department of Mechatronics Engineering, Wah Engineering College, University of Wah, Wah, Pakistan

[8] Department of Electrical Engineering, Riphah International University, Islamabad, Pakistan

[9] School of Computing, Edinburgh Napier University, Edinburgh, UK