Ain Shams Engineering Journal 15 (2024) 102507

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Design and optimization of nonlinear component of block cipher: Applications to multimedia security

Adil Waheed [a], Fazli Subhan [a,b], Mazliham Mohd Suud [b,*], Muhammad Mansoor Alam [b,c], Sajjad Haider [a]

[a] Faculty of Engineering and Computer Sciences, National University of Modern Languages, Islamabad, Pakistan
[b] Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia
[c] School of Computer Science in the Faculty of Engineering and IT, University of Technology Sydney, Australia

ARTICLE INFO

Keywords:
S-box
Nonlinear component
Image encryption
Nonlinearity
Security
Chaotic maps

ABSTRACT

Nowadays, various image encryption schemes based on chaotic systems have been developed, each of them has its own limitations and strength in terms of security and computational speed. The proposed image encryption scheme utilizes 2-D maps without disturbing their mathematical structure, characterized by topological features such as chaotic behavior and fractal properties, namely the Zaslavasky, Bakers, and Henon Maps. This approach utilizes both confusion and diffusion stages to achieve high levels of security against various attacks. The confusion stage utilizes chaotic values to muddle the rows and columns of the image, reducing the correlations between neighboring pixels, while the diffusion step achieves the avalanche effect with 2D Bakers map and Henon map. The proposed image encryption scheme is analyzed thoroughly to evaluate its security and performance. To evaluate the security and computational efficiency of the proposed image encryption method, various analysis such as correlation, contrast, entropy, energy, homogeneity, and performance analyses are conducted. Moreover, the three proposed S-boxes are also tested to evaluate their effectiveness using cryptographic analysis tests such as nonlinearity, strict Avalanche criterion, differential probability, linear probability, and bit independence criterion, which we also utilized in our proposed image encryption scheme.

1. Introduction

The S-box is a fundamental component of a modern block cipher and plays a crucial role in ensuring the security of encrypted communications. S-boxes are designed to provide nonlinear substitution of input data, which helps to dubious the relationship between the plaintext and ciphertext. The S-box is the only nonlinear component of AES, which is one of the most conventional ciphers. Mostly conventional ciphers are not best suited for image encryption due to the unique properties and characteristics of image data [1,2]. Image encryption is entirely different from encryption of textual data because textual data has higher redundancy, which means that encryption algorithms can take advantage of patterns and repetitions. Wang et al. [3] proposed an image encryption scheme based on a laser chaotic system and XOR diffusion, respectively, to achieve rapid transmission and reduce the statistical characteristics of the plain image. The researchers in [4] investigated a color image cryptosystem based on multiple layers fractional-order hyper chaotic map and pseudo random number generator (PRNGs). In [5], a detailed analytical review regarding construction of nonlinear component is presented. An efficient image encryption algorithm was studied in [6] that combines the shuffling of pixel positions and altering the grayscale values of image pixels to create confusion. By employing PRNG bitstreams generated from cellular automata and the Lorenz system as keys, along with the integration of the S-box, the encryption process achieves well-secured images [7]. The use of PRNGs in image cryptosystems is well-documented in numerous literature examples. The authors of [8] introduced TRNG (True Random Number Generator) based image encryption that relies on the stochastic duration time of double threshold-switching (TS) memristors.

In recent years, there has been a growing interest in using chaotic maps for image encryption due to their ability to produce complex and random sequences that are difficult to predict [9]. Chaotic maps are mathematical functions that exhibit sensitive dependence on initial conditions, which makes them useful for generating random numbers and creating nonlinear transformations that can be used to strengthen cryptographic algorithms [10]. This technique [11] utilizes Arnold's cat

* Corresponding author.

https://doi.org/10.1016/j.asej.2023.102507
Received 9 July 2023; Received in revised form 5 August 2023; Accepted 16 September 2023
Available online 7 October 2023

Table 1
Proposed Chaotic S-box based on modified Zaslavsky May.

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

map for shuffling of image and also, used cyclic chaos and PRNG as an additional security measure. Chaotic systems have been used for image encryption because they can generate a large amount of random data which can be used as encryption keys or as part of the encryption process. Chaotic systems are also attractive because they are relatively easy to implement and can produce complex and unpredictable behavior. In chaotic image encryption systems, the positions or values of the pixels in the original image are usually exchanged using a chaotic stream cipher. The chaotic stream cipher generates a sequence of random numbers that are used to shuffle the pixels in the image or to modify their values in some way. This process makes it difficult for an attacker to decipher the original image without knowing the encryption key. By incorporating chaotic maps into vulnerable cryptographic schemes, researchers hope to improve their resistance to attacks. Likewise the authors of [12] propose a novel framework for image encryption with the help of two hyperchaotic maps and the single neuron model (SNM). In [13], authors utilized a coset graph to design an S-box with better cryptographic characteristics. Wang et al.[14] presented big data related architecture and general framework which has an ability to collect and store data, including processing, analysis, performance, and security for big data applications. In [15], a comprehensive analysis of blockchain-based applications in the context of security and privacy is presented. Features of gray image are extracted in high frequency domain using multi-resolution multi-direction filtering [16]. This paper [17] presents an optimized approach for cloud provider and its users with better request allocation strategy to reduce energy cost. Lu et al. [18] introduced coupled images decomposition algorithms. These algorithms are robust and accurately restore the images.

In this paper, we propose chaotic based S-boxes using 2-D maps i.e. Zaslavasky map, Honen map and Bakers map. The Zaslavasky is a nonlinear, dynamical and discrete-time system. It produces dynamic and deterministic behavior which represents an essential part of the encryption algorithms. Moreover, this map is very important it is sensitive to control parameters and can be utilized to create a pseudo-random number generator during the permutation phase. This unique characteristic of sensitivity to control parameters makes it a valuable tool for various cryptographic applications that necessitate randomness [19]. We also propose an S-box using Hénon map. The Hénon map is a two-dimensional discrete-time dynamical system that was introduced by Michel Hénon. It is a simple nonlinear map that exhibits chaotic behavior for certain parameter values. This chaotic behavior can be used in various applications, including image encryption, random number generation, and secure communications. In addition, we constructed an S-box based Baker's map. Currently, Baker's map is being used as a basis for secure communication in stream ciphers. Furthermore, we propose an innovative image encryption algorithm that incorporates each of the proposed S-boxes.

2. Preliminaries

The proposed work is based on using 2D chaotic maps, which are mathematical representations of complex systems that exhibit unpredictable and non-repeating behavior for communication of secure data. We drew inspiration from this source [20–22] and acquired knowledge of these algorithms as foundational concepts. This section briefly reviews three chaotic maps, namely the Zaslavasky map, Henon map, and Baker map. Zaslavasky map [21] is a two-dimensional chaotic map that is being used in cryptography. A set of nonlinear equations (1 & 2) define the map, and even for slight modifications to its initial conditions, it behaves in a complex and unpredictable manner. In order to create pseudorandom numbers and create cryptographic algorithms like S-boxes, the Zaslavasky map has been used in different research areas

Table 2
S-box derived from modified Henon map.

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

Table 3
S-box construction based on modified Baker's Map.

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

Table 4
Experimental Results and Comparison with known S-boxes.

S-box	Nonlinearity			SAC	BIC-NL	BIC-SAC	DP	LP
	MaxVal	MinVal	AvrVal	AvrVal	AvrVal	AvrVal	MaxVal	MaxVal
Abd-El-Atty [26]	106	98	101.50	0.5043	104.28	0.5071	12	0.125
Zhu [27]	108	102	105.75	0.5021	104.14	0.5050	10	0.1328
Zahid [28]	112	110	111.75	0.5029	103.74	0.5005	10	0.125
Alghafis [29]	105	97	102.87	0.5192	102.67	0.4787	54	0.1679
Jiang [30]	108	104	106.75	0.4975	103.57	0.5022	10	0.1328
Dimitrov [31]	114	116	114.50	0.5012	104.21	0.5046	10	0.1406
Shahzad [32]	112	108	110.50	0.5031	109.21	0.5018	6	0.0859
Hua [33]	108	102	105.25	0.5351	103.25	0.5087	–	0.1406
Javeed [34]	110	106	107.50	0.4997	104.64	0.5048	12	0.1406
Ali [35]	106	98	102.75	0.4992	103.07	0.0140	12	0.1406
Hematpour [36]	109	102	105.25	0.4960	104.53	0.5045	–	0.1328
Nizam Chew [37]	112	112	112.00	0.4980	112.00	0.4981	4	0.0625
Artuğer [38]	112	110	111.75	0.4968	104.00	0.5016	12	0.125
Proposed S-box-1	**116**	**116**	**113.75**	**0.4956**	**103.42**	**0.4992**	**10**	**0.1406**
Proposed S-box-2	**112**	**116**	**114.25**	**0.4978**	**103.41**	**0.5044**	**10**	**0.125**
Proposed S-box-3	**112**	**112**	**112**	**0.5063**	**103.86**	**0.4987**	**10**	**0.1328**

[23]. Henon map [22] is a two dimensional map that exibits chaotic behavior. The Henon map has been widely used as building block for cryptographic primitives due its chaotic behavior, such as construction of nonlinear component of block cipher and stream ciphers [24,25]. It is defined by following two nonlinear equations (Eq. 3 & 4) and maps each point in a 2-D plane to a new point. Whereas, the Baker's is also a 2-D map [20] that has been used in cryptography, image processing, signal processing, and encryption applications because of its sensitivity to initial conditions and its capacity to generate pseudo random numbers. The Baker's map is used to map square grid of points onto itself. It is described by a set of recurrence equations that iteratively transform the coordinates of every grid point.

3. Proposed S-box scheme

In this article, we propose a novel S-box design and image encryption scheme that utilize chaotic maps, specifically the Zaslavas, Bakers, and Henon maps. These maps are known for their chaotic behavior and sensitivity to initial conditions, which make them useful for generating pseudo-random sequences that can be used in cryptographic applications. Our generated S-boxes meet all relevant standards and exhibit impressive results in analysis (see Tables 1–4), including nonlinearity (NL), Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), LP (linear probability), and DP (Differential probability).

3.1. Modified Zaslavsky map, Henon Map, and Baker's map based nonlinear component construction without disturbing their mathematica structure

The proposed scheme of the Zaslavsky map algorithm involves a series of well-defined steps to generate pseudorandom numbers (PRNs). The process starts with initializing the variables to some predetermined values, which are then used to calculate the parameter μ.

Once μ is determined, a loop is initiated to perform a 2 million (2,000,000) iterations. Within this loop, the Zaslavsky map equations are used to generate the new values of x and y, which are then used to calculate four PRNs. These PRNs are printed in decimal and binary form, and the values of x and y are updated to their new values for the next iteration of the loop. The algorithm can be repeated with variations to generate longer sequences of PRNs. The strength of this algorithm lies in its chaotic nature, which makes it difficult for an adversary to predict the sequence of PRNs generated.

The following is an algorithm based on Zaslavasky map for generating resistant to cryptanalysis nonlinear component of block Cipher.

Algorithm-1: Proposed S-box construction Algorithm-1

Step 1. Initialize variables x, y, v, e, τ and size with some specific required values.
Step 2. Calculate the value of μusing the formula $1-e^{-\tau}/\tau$.
Step 3. While (size<=2,000,000).
Step 3.1. Calculate the new values of x_1 and y_1 using the chaotic map equations:
$x_1 = mod(y_n + v(1+\mu z_n) + \epsilon v \mu z_n \cos(2\pi y_n), 1)$
$y_1 = e^{-\tau}(z_n + \cos(2\pi y_n))$
Step 3.2. Calculate pseudo random number for S-box design

(*continued on next page*)

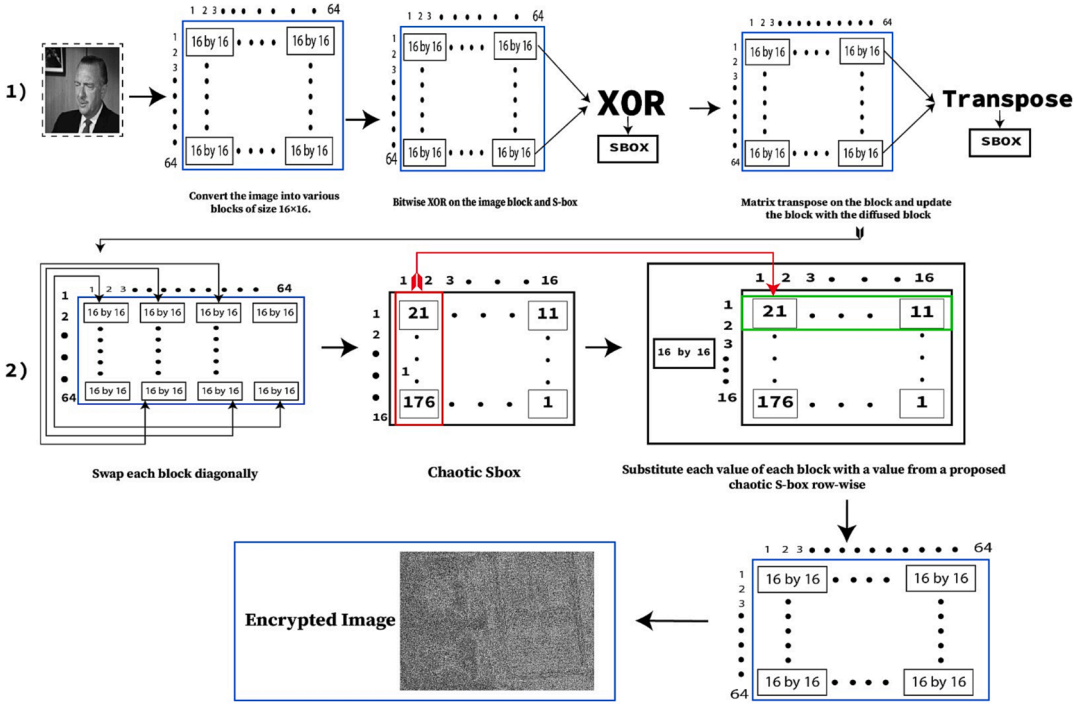

Fig. 1. A pictorial view of an image encryption algorithm.

(*continued*)

Algorithm-1: Proposed S-box construction Algorithm-1

$prn = mod(floor(x_1 \times (2^{42-1})), 256)$
$binaryNo = convert(prn)$ ▷ This function will convert the PRN into binary number
$firstComp = complementCal(binaryNo)$ ▷ The function calculates first complement
XORingResult←ApplyXORbetweenfirxtCompande
SboxArray←XORedvaluedisstoredintheSboxArrayatthecorresponingindex
Step 4. End of while loop
Step 7. To construct reliable S-box, apply permutation to the newly generated pseudo random numbers.

(*continued*)

Algorithm 2: Proposed S-box design algoritm-2

Step 3.4.3. For each position, flip the binary representation.
Step 3.4.4. Reverse the binary representation and return the result in decimal form.
Step 3.4.5. Change the bits in the original position to match the bits in the reversed position.
Step 3.4.6. Store the resultant value into SboxArray.
Step 4. End of ForEach loop
Step 9. Organize the resulting values into a 16 × 16S-box for cryptosystems.

The proposed algorithm below commences by setting the initial values of x and y, as well as the values for a, b, and the total number of iterations to be carried out. The Henon map is iterated 1.5 million times (1,500,000) to generate unpredictable values of x and y, which are utilized to calculate a new value of x. This value is then multiplied by a specific value and rounded down to produce a pseudorandom number. This random number is constrained to a value between 0 and 255 by computing its modulus with 256. This process is repeated iteratively until we get desired bijective values, and the resulting values are organized into a 16 × 16S-box.

The following is an algorithm to generate secure S-box using Henon Map:

Algorithm 2: Proposed S-box design algoritm-2

Step 1. Set the initial values of x and y.
Step 2. Set the values of a, b, and size.
Step 3. FOREACH 1:1,500,000.
$x_{n+1} = 1 + y_n - ax_n^2$
$Y_{n+1} = bx_n$
Step 3.1. Print the chaotic values of x_1 andy_1.
Step 3.2. For each new value of x_1, multiply it by (2^{35-1}) and floor it.
$prn = (floor(x_1 \times (2^{35-1})))$
Step 3.3. Take the modulus of pseudorandom number with 256 to get a value between 0 and 255.
Step 3.4. Apply following bit shuffling permutation on PRN
Step 3.4.1. convert PRN into binary
Step 3.4.1. Take the binary sequence of length N
Step 3.4.2. Represent each bit's position in binary form.

(*continued on next column*)

The Baker's map design an S-box by defining the parameters $a, size, x, y$, and required arras. Subsequently, unpredictable values of *xandy* are generated by Henon map. The two different branches of the this map modify the values *x*and *y*. These modified values are stored into *xSboxkeyandySboxkey* arrays, respectively. Chaotic values of S-box are actually floating point numbers. These numbers are multiplied with any specific number and take modulo with 256 to constraint the values in range of 0 to 255. The following is a procedure to construct an S-box using Baker's map.

Algorithm 3: S-box derived from modified Baker's map

Step 1. Define the parameters and initializes required arrays with zeros.
Step 2. For Loop from 1 to rang (2,000,000)
Step 2.1. If 0 <= x <= 0.5 ←update × and y according to the first branch of the map
x ← 2 * x
y ← a * y
Step 2.2. If 0.5 < x <= 1 ←update × and y according to the second branch of the map
x ← 2 * x − 1,
y ← a * y + 0.5
Step 6. Store the updated values of x and y into *xSboxkeyandySboxkey* arrays, respectively.
Step
Step 7. Multiply each value of *xSboxkey*arrays with specific number then take mod with 256 to constraint values between 0 and 255.
Step 8. Convert updated *xSboxkey* into corresponding 8-bit binary number.
Step 9. Define 8-bit binary constant (key) for addition-based hash permutation.
Step 10. Add the updated *xSboxkey* and the key together (modulo 256).
Step 11. Store the resultant value into SboxArray

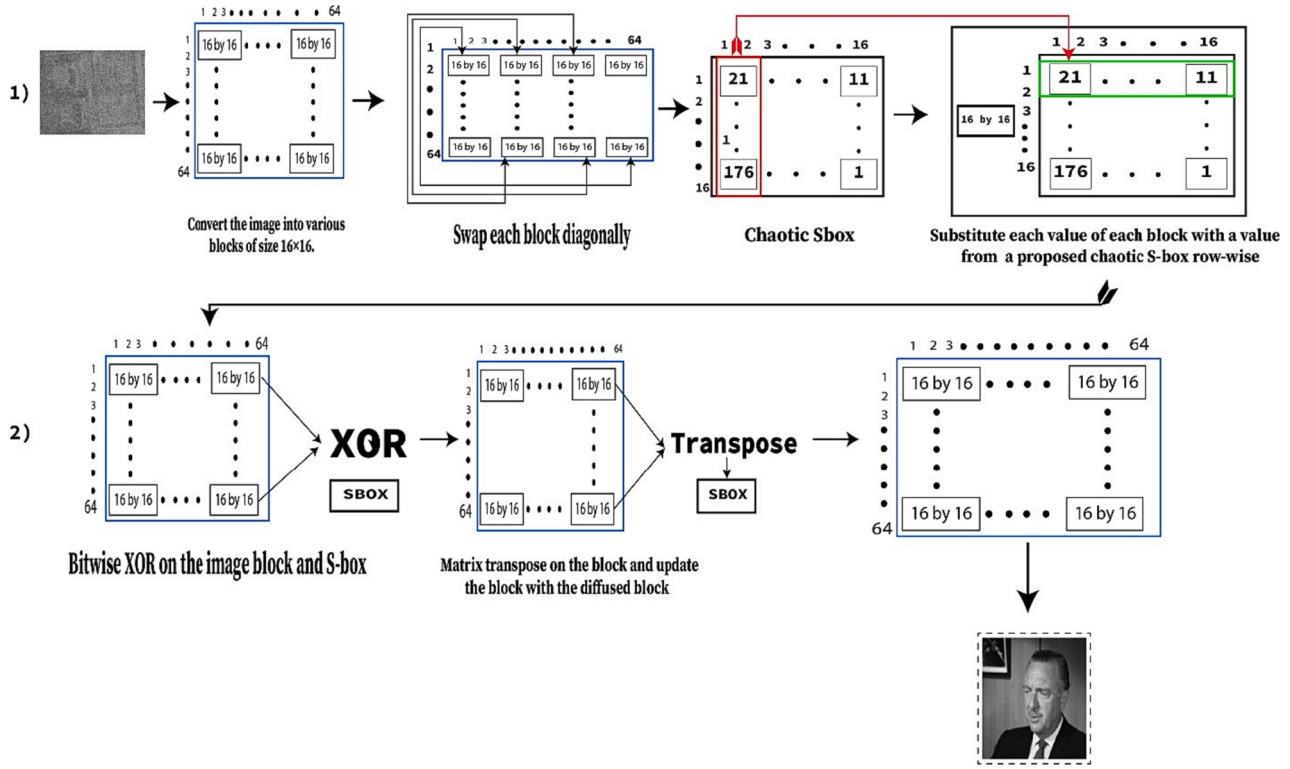

Fig. 2. A pictorial view of an image decryption algorithm.

4. Proposed image encryption algorithm

Both one dimensional (1-D) and two dimensional (2-D) maps are used in the field of cryptography. Generally, 1-D maps are used to generate encryption keys or digital signatures but the issue is that generated keys with 1-D maps are small in size and have lower degree of complexity and randomness. However, 2-D maps are considered more secure due to larger key size, parameters, space, and complexity. That's the reason we employed 2-D maps (Zaslavasky, Henon, Bakers). These maps reached a chaotic state to provide nonlinear and dynamical behavior.

In this image encryption algorithm, initially, an assessment is conducted to determine whether the image is grayscale or RGB. In case of RGB, the image is transformed into a grayscale. Next, a verification is made to determine whether the dimensions of the image are divisible by 16, if it is not then in this case, pad the image with some extra pixels on the sides until the width and height are both divisible by 16. The following is an algorithm to solve perfect division issue:

- Read and calculate the actual height and width of the image.
- Find the multiples of 16 that are closest (greater) to or equal to the current height and width. Assign the names 'new_height' and 'new_width'to these new values.
- Subtract the actual height and width from 'new_height' and 'new_width'
- Create the blank canvas with new_height and new_width.
- Copy the original image to the center of canvas
- Padding the remaining area with white pixels.
- The padding image is perfectly divisible by 16.

Once the image has been subdivided into 16 × 16 blocks, a swapping process is conducted whereby the upper diagonal of the first block is swapped with the lower diagonal of the last block, the lower diagonal of first block is swapped with upper diagonal of last block and vice versa. This process is repeated for each block in the image. After the completion of the swapping process, perform row-wise substitution on each value of each block with value of proposed chaotic S-box to encrypt the image (See Fig. 1). The following is the novel image encryption algorithm based on the proposed chaotic S-boxes.

Algorithm 4: Proposed Image Encryption Scheme

Step 1. Read the original image and examine its dimensions.
Step 2. Check whether the image is Grayscale or RGB.
Step 3. If the image is RGB, convert it into a Grayscale image.
Step 4. Check if the image is divisible into 16 × 16 blocks.
Step 5. If the image is not divisible into 16 × 16 blocks, add padding (white pixels) to make an image size divisible by 16.
Step 6. Convert the image into various blocks of size 16 × 16. Each block individually reveals no information about the plain image.
Step 7. Apply a confusion layer to each block, i.e., perform a bitwise XOR on the image block and S-box and update the block with the confused block.
Step 8. Apply a diffusion layer to each block i.e., perform matrix transpose on the block and update the block with the diffused block.
Step 9. Perform multiple rounds of the confusion layer and diffusion layer to enhance the encryption strength and ensure better dispersion of pixel values.
Step 10. Swap each block diagonally to further muddle the number of rows and columns and introduce additional confusion.
Step 11. Substitute each value of each block with a value from a proposed chaotic S-box row-wise, using a lookup operation based on the S-box table.
Step 12. Repeat Steps 7 to 11 for a desired number of rounds.
Step 13. Finally, the image is encrypted, and the encrypted image is obtained.

Additionally, we selected an image block of size 16x16 due to following reasons:

- In our proposed image encryption algorithm, we use an Substitution Box (S-box) of size 16 × 16, which is a well-known and standard size.
- The image block of size 16 × 16 provides a good balance between cryptographic strength and computational efficiency. This size guarantees robust nonlinearity, resistance to linear and differential cryptanalysis, and compatibility with modern cryptographic algorithms.

Table 5
Nonlinearity for Proposed S-boxes.

Proposed S-boxes	Nonlinearity:							
	0	1	2	3	4	5	6	7
Proposed S-box-1	114	114	114	112	112	116	112	116
	Average Nonlinearity: 113.75							
Proposed S-box-2	112	114	114	114	114	116	114	116
	Average Nonlinearity: 114.25							
Proposed S-box-3	112	112	112	112	112	112	112	112
	Average Nonlinearity: 112							

4.1. Image decryption algorithm

In order to decrypt the image, each step must be carried out in reverse order, exactly as it was done during the encryption process. This process ensures that original image is recreated perfectly (see Fig. 2). Here are the steps to decrypt an encrypted image.

Algorithm 5: Image Decryption Scheme

Step 1. Read the encrypted image and examine its dimensions.
Step 2. Reverse the row-wise substitution process by swapping each entry of each block with the chaotic S-box.
Step 3. Diagonally swap each block of the encrypted image.
Step 4. To reverse the diffusion process, transpose each block back to its original form.
Step 5. Undo the confusion operation by performing XOR between each block of encrypted image and chaotic S-box.
Step 5. To recreate the original image, reassemble the decrypted blocks back together.
Step 6. To get the actual dimensions of the decrypted image, remove the zero padding.
Step 7. If the original image was converted from RGB to grayscale, the decrypted image is restored to its original color format.
Step 9. Finally, the image is decrypted back to its original form.

5. Performance analysis of proposed chaotic S-boxes

In order to analyze the cryptographic properties of proposed chaotic S-boxes, we perform bijectivity, Strict Avalanche criterion (SAC), nonlinearity (NL), BIC (Bit Independence Criterion), linear probability (LP), and Differential probability (DP) measurement criteria. We also examine resistance of chaotic S-boxes against linear and differential attacks.

5.1. Nonlinearity

Linearity is viewed as a curse in the field of cryptography because it makes system vulnerable to attacks. In contrast, nonlinearity ensures that the output of cryptographic system is not a linear combination of its vectors. If an S-box has linear mapping between its input and output, its resistance level is very low [39]. So, nonlinearity is a metric to assess the level of confusion and diffusion provided by the S-box. It refers to the degree to which the output bits of an S-box (nonlinear component of block cipher) are dependent on its input bits. In other words, a high degree of nonlinearity makes it challenging for an attacker to predict the S-box's output from its input. AES S-box has nonlinearity 112 which provides good resistance against linear and differential attacks. It is pertinent to mention that nonlinearity results of our proposed S-boxes are better than AES, as shown in Table 5 below.

$$N_\theta = 2^{n-1} - \frac{1}{2}[\left(h \in GF(2)^{n^{max}}\right)|S_\theta(h)|] \tag{6}$$

$$S_\theta(h) = \sum_{g \in GF(2)^n} (-1)^{\theta(g) \oplus \mathrm{g.h}} \tag{7}$$

5.2. Strict avalanche criterion

The strict avalanche criterion (SAC) measures the sensitivity of S-box to small possible change in its input. This criterion quantifies how much the output of an S-box changes when a single input bit is flipped. The result of an S-box satisfying this property is that each output bit will change with a probability that is close to 0.5 [40], so this value examine that the SAC is suitable property for measuring the sensitivity of S-box. We examine our three proposed S-boxes as shown in (see Tables 6–8). The average results of SAC are near to 0.5, which indicates that criterion is satisfied .

5.3. Bit independence criteria (BIC)

BIC is an another criterion to evaluate the cryptographic properties of an S-box. BIC is a metric used to assess how statistically output bits of an S-box are independent from its input bits [41]. A BIC value that is close to zero means that the S-box has strong bit independence and thus offers good diffusion. In case the value of BIC is significantly greater than zero, which indicates S-box is not statistically independent and may be vulnerable to attacks. Here are the BIC results Tables 9–11 of the proposed S-boxes while comparing them with well-known S-boxes.

5.4. Differential approximation probability

The differential approximation probability (DP_f) refers that a Boolean function will produce a particular value in response to a change in its input. This change represents XOR distribution between input and output. The input is represented is by Δx and the output is denoted as Δy. The formula to calculate differential probability is given as follows:

$$DP_f = max_{\Delta x \neq 0, \Delta y}\left(\frac{\{x \in X | f(x) \oplus \mathrm{f(x} \oplus \Delta \mathrm{x} = \Delta \mathrm{y}\}}{2^n}\right) \tag{8}$$

Where 2^n represents the number of elements in the set, X represents a set of all possible inputs. The value of DP_f affects how well the S-box defends against differential attacks. The S-box is regarded as having strong defenses against these attacks if $DP_f$is small. On the other hand, if $DP_f$is large, it is thought that the S-box is incapable of fending off such attacks. The following are the differential probability results for our proposed S-boxes (see Table 12).

Table 6
SAC Results for S-box-1.

Strict Avalanche Criterion:								
SAC for Proposed S-box-1	0	1	2	3	4	5	6	7
	0.59375	0.48438	0.46875	0.53125	0.54688	0.51563	0.46875	0.50000
	0.48438	0.51563	0.50000	0.53125	0.46875	0.48438	0.46875	0.53125
	0.48438	0.48438	0.48438	0.45313	0.45313	0.46875	0.48438	0.50000
	0.50000	0.59375	0.54688	0.50000	0.51563	0.46875	0.53125	0.42188
	0.45313	0.42188	0.51563	0.46875	0.43750	0.51563	0.56250	0.46875
	0.46875	0.50000	0.46875	0.53125	0.48438	0.45313	0.48438	0.53125
	0.48438	0.45313	0.53125	0.51563	0.50000	0.53125	0.46875	0.46875
	0.50000	0.50000	0.46875	0.51563	0.50000	0.50000	0.53125	0.50000
	Average SAC: 0.49560							

Table 7
SAC results for S-box-2.

SAC for Proposed S-box-2	0	1	2	3	4	5	6	7
	0.57813	0.48438	0.54688	0.48438	0.56250	0.50000	0.46875	0.45313
	0.50000	0.50000	0.56250	0.56250	0.51563	0.53125	0.46875	0.50000
	0.43750	0.48438	0.46875	0.46875	0.51563	0.48438	0.51563	0.46875
	0.46875	0.51563	0.43750	0.43750	0.48438	0.51563	0.51563	0.56250
	0.54688	0.50000	0.46875	0.48438	0.56250	0.51563	0.53125	0.46875
	0.48438	0.45313	0.48438	0.53125	0.46875	0.50000	0.46875	0.53125
	0.48438	0.45313	0.45313	0.45313	0.53125	0.46875	0.46875	0.53125
	0.50000	0.50000	0.53125	0.50000	0.50000	0.50000	0.46875	0.51563
	Average SAC: 0.49780							

Table 8
SAC Results for S-box-3.

SAC Proposed S-box-3	0	1	2	3	4	5	6	7
	0.46875	0.50000	0.45313	0.50000	0.57813	0.48438	0.53125	0.56250
	0.50000	0.48438	0.46875	0.45313	0.48438	0.50000	0.51563	0.54688
	0.46875	0.56250	0.46875	0.50000	0.53125	0.48438	0.51563	0.42188
	0.50000	0.48438	0.54688	0.43750	0.50000	0.57813	0.57813	0.56250
	0.48438	0.51563	0.46875	0.50000	0.48438	0.51563	0.53125	0.53125
	0.54688	0.45313	0.48438	0.53125	0.56250	0.54688	0.50000	0.45313
	0.54688	0.45313	0.54688	0.51563	0.50000	0.50000	0.57813	0.50000
	0.46875	0.51563	0.48438	0.54688	0.50000	0.45313	0.51563	0.50000
	Average SAC: 0.50634							

Table 9
BIC for Proposed S-box-1.

BIC for Proposed S-box-1	0	1	2	3	4	5	6	7
	——	0.505859	0.511719	0.503906	0.507812	0.486328	0.492188	0.492188
	0.505859	——	0.503906	0.505859	0.490234	0.523438	0.490234	0.496094
	0.511719	0.503906	——	0.513672	0.474609	0.505859	0.490234	0.496094
	0.503906	0.505859	0.513672	——	0.517578	0.494141	0.478516	0.513672
	0.507812	0.490234	0.474609	0.517578	——	0.490234	0.494141	0.500000
	0.486328	0.523438	0.505859	0.494141	0.490234	——	0.498047	0.500000
	0.492188	0.490234	0.490234	0.478516	0.494141	0.498047	——	0.501953
	0.492188	0.496094	0.496094	0.513672	0.500000	0.500000	0.501953	——
	Average BIC: 0.49923							

Table 10
BIC for Proposed S-box-2.

BIC for Proposed S-box-2	0	1	2	3	4	5	6	7
	——	0.498047	0.519531	0.507812	0.500000	0.494141	0.503906	0.488281
	0.498047	——	0.511719	0.492188	0.529297	0.527344	0.507812	0.507812
	0.519531	0.511719	——	0.494141	0.496094	0.501953	0.496094	0.498047
	0.507812	0.492188	0.494141	——	0.513672	0.507812	0.496094	0.515625
	0.500000	0.529297	0.496094	0.513672	——	0.496094	0.519531	0.494141
	0.494141	0.527344	0.501953	0.507812	0.496094	——	0.492188	0.500000
	0.503906	0.507812	0.496094	0.496094	0.519531	0.492188	——	0.515625
	0.488281	0.507812	0.498047	0.515625	0.494141	0.500000	0.515625	——
	Average BIC: 0.01026							

Table 11
BIC for Proposed S-box-3.

BIC for Proposed S-box-3	——	0.484375	0.478516	0.5	0.503906	0.494141	0.519531	0.513672
	0.484375	——	0.496094	0.505859	0.503906	0.482422	0.492188	0.496094
	0.478516	0.496094	——	0.521484	0.503906	0.501953	0.474609	0.480469
	0.5	0.505859	0.521484	——	0.478516	0.507812	0.521484	0.523438
	0.503906	0.503906	0.503906	0.478516	——	0.490234	0.501953	0.498047
	0.494141	0.482422	0.501953	0.507812	0.490234	——	0.498047	0.490234
	0.519531	0.492188	0.474609	0.521484	0.501953	0.498047	——	0.5
	0.513672	0.496094	0.480469	0.523438	0.498047	0.490234	0.5	——
	Average BIC: 0.01256							

Table 12
DP for Proposed S-boxes.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	6	6	8	8	6	6	8	6	6	8	10	8	6	6	6
8	8	8	8	8	6	6	8	6	6	6	6	6	8	8	6
8	6	6	6	8	6	6	8	6	6	6	6	8	8	6	6
8	8	8	8	8	6	6	6	6	6	8	6	6	6	6	6
6	8	8	6	8	6	8	8	6	6	6	8	8	6	8	6
6	8	6	8	6	6	8	8	8	6	10	8	6	6	6	6
6	6	8	6	6	8	4	6	8	8	6	8	6	8	8	6
10	8	6	6	6	6	8	8	8	8	8	8	6	8	6	10
6	6	8	8	10	8	8	6	8	6	8	6	6	8	10	6
6	8	6	6	6	8	6	8	8	6	6	6	6	6	8	8
6	8	6	6	8	8	6	10	6	6	8	8	6	6	6	8
6	6	6	6	8	6	8	6	6	6	6	8	6	6	6	8
6	6	6	6	6	10	8	8	10	6	6	4	6	6	8	6
6	6	6	6	6	6	6	6	6	8	8	6	6	6	8	10
8	6	10	6	6	6	6	6	8	6	8	6	8	6	6	8
6	8	6	6	8	6	8	8	6	6	6	8	6	6	8	6

Max DP Value for Proposed S-box-1: 10
Max DP Value for Proposed S-box-2: 10
Max DP Value for Proposed S-box-3: 10

Table 13
LP for Proposed S-boxes.

Linear Approximation Probability:	Max Count	Max Value
LP for proposed S-box-1	164	0.140625
LP for proposed S-box-2	160	0.125
LP for proposed S-box-3	162	0.13281

5.5. Linear approximation probability

Linear approximation probability (LP) is one of the most important tests in order to design and analysis of S-boxes. An S-box's resistance to linear cryptanalysis attacks is gauged by its LP. The S-box is less vulnerable to this kind of attack if the LP value is higher. As a result, it is essential to design S-boxes with high values for LP in order to ensure the security of cryptographic systems (see Table 13). The total number of input and output bits that are coincident is used to calculate LP. The following is the mathematical formula to calculate the LP.

$$LP_{(a,b)} = (2^{n-1}+1)^{-1} \times |\{x : S(x)S(xa) = b\}| \quad (9)$$

Where a and b are input and output differences respectively and n is a bit-length of the input and output.

5.6. Bijectivity

Bijectivity is the capacity for each input value to be specifically mapped to a specific output value and vice versa. The S-box's bijectivity is essential for the security of the encryption algorithm because it guarantees that the encryption process is reversible and that no data is lost. Our proposed S-boxes are are bijective when their nonlinearity results

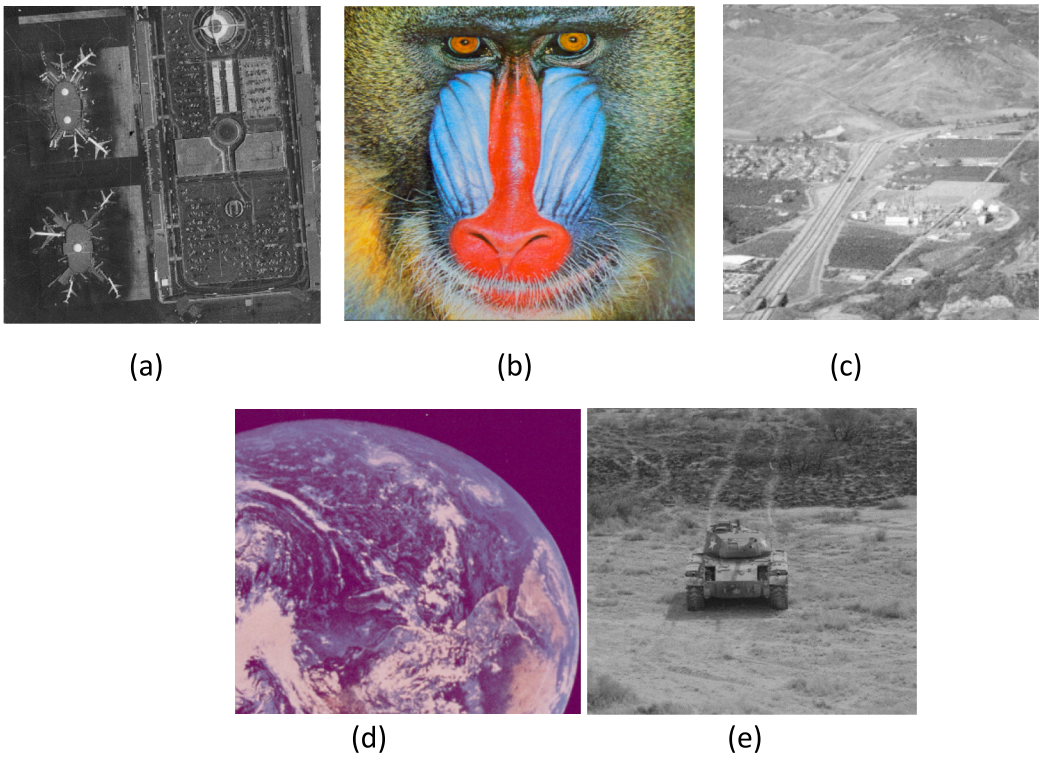

Fig. 3. Set of plain images (a-e) of Airport, Baboon, chemical plant, earth space, and Tank to examine image encryption algorithm.

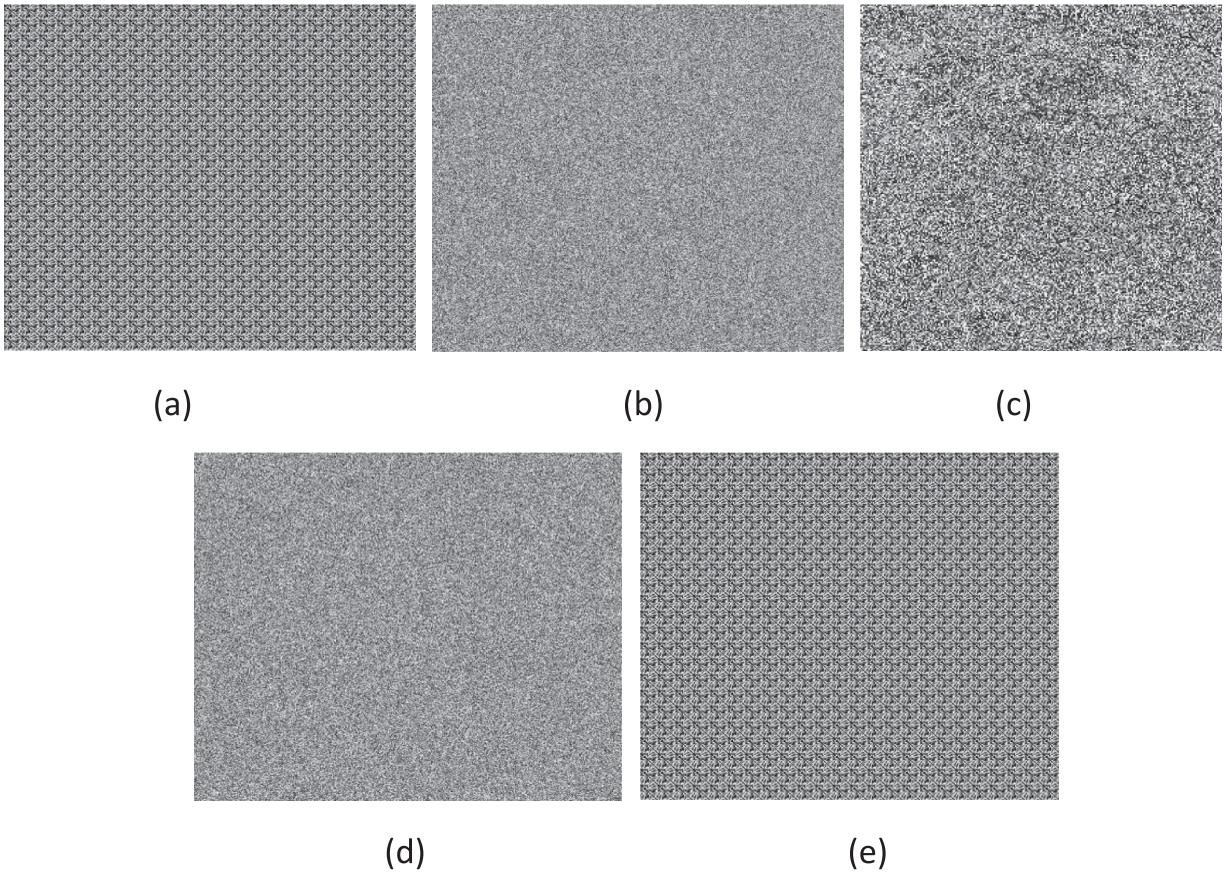

Fig. 4. Set of Encrypted images (a-e) of Airport, Baboon, chemical plant, earth space, and Tank.

are greater than or equal to 112.

6. Analysis of cryptographic properties of an encrypted image

The performance and security attributes of the proposed chaotic-based S-box encryption scheme is assessed through a number of experimental tests. These tests are set up and run on a Windows 10 desktop with a Core i5 processor, 8 GB of RAM, and the MATLAB 2020a platform. Standard RGB and grayscale images (airport, a baboon, a chemical plant, an earthly space, and a tank) are included in the experiment, which are chosen from the USC-SIPI image dataset. Figs. 3 and 4 provide an illustration of these experimental images .

6.1. Correlation analysis

Images may be subject to security breaches during transmission or storage due to the high degree of correlation between adjacent pixels. As a result, encryption techniques for images therefore aim to reduce this correlation to conceal image content. It measures the correlation between adjacent pixels and has a range of −1 to 1. No correlation is indicated by a value of 0, maximum positive correlation is represented by a value of 1, and maximum negative correlation is indicated by a value of −1. The correlation coefficients between adjacent pixels in a cipher image can be minimized to produce low correlation values, which denote higher security.

The correlation of adjacent pixels can be calculated as:

$$CC_v = \frac{\sum_{i=1}^{H-1}\sum_{j=1}^{w}(C_{(i,j)} - \overline{C})(C_{(i+1,j)} - \overline{C})}{\sqrt{\sum_{i=1}^{H-1}\sum_{j=1}^{w}(C_{(i,j)} - \overline{C})^2 \sum_{i=1}^{H-1}\sum_{j=1}^{w}(C_{(i+1,j)} - \overline{C})^2}} \quad (9)$$

This equation presented in this context employs the width and height of an image as W and H, respectively. The adjacent pixel values of the cipher image at positions $(i+1,j)$ and $(i,j)$are represented by C (i + 1,j) and C (i,j), respectively. The below mathematical expression is used to calculate the correlation coefficient between plaintext and its corresponding cipher image:

$$CC_{P,C} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{w}(P_{(i,j)} - \overline{P})(C_{(i,j)} - \overline{C})}{\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{w}(P_{(i,j)} - \overline{P})^2 \sum_{i=1}^{H}\sum_{j=1}^{w}(C_{(i,j)} - \overline{C})^2}} \quad (10)$$

This equation uses the width and height of the images, which are denoted here by the letters W and H, respectively. According to Table 14, the proposed algorithm exhibits low correlation values between the plaintext and cipher images. This implies a high level of security because it suggests there is little similarity or predictability between the original message and the encrypted image.

Table 14
The correlation coefficients of different images.

Correlation Coefficient				
Image	Size	Vertical	Horizontal	Diagonal
1. Airport	1024 × 1024	0.037239	0.039534	0.072504
2. Baboon	512 × 512	0.0036625	0.0030674	−0.0017916
3. Chemical Plant	256 × 256	0.0030192	0.0045205	0.0051687
4. Earth Space	512 × 512	0.0010803	0.0036398	−0.00012869
5. Tank	512 × 512	0.089371	0.017622	0.0088644

6.2. Entropy analysis

Entropy is used to evaluate the degree of uncertainty and randomness in image. It is widely used in communication system to specify the

Table 15
Entropy Results.

Entropy Analysis		
Image	Color and Size	Entropy
Airport	1024 × 1024	7.9
Baboon	512 × 512	7.7
Chemical Plant	256 × 256	7.3
Earth Space	512 × 512	7.7
Tank	512 × 512	7.8

Table 16
Homogeneity Results.

Homogeneity		
Image	Color and Size	Homogeneity
Airport	1024 × 1024	0.37297
Baboon	512 × 512	0.46778
Chemical Plant	256 × 256	0.40667
Earth Space	512 × 512	0.46884
Tank	512 × 512	0.37137

amount of information in a message. The probability distribution of the pixel values in the encrypted image can be used to estimate the entropy. A higher level of randomness among pixels indicates that the image is more secure against attacks. Mathematically it is represented as:

$$H(k) = -\sum_{i=0}^{2n-1} p(k_i)\log_2 p(k_i) \tag{11}$$

In this Equ. (11) H represent entropy which measure in bits, and $p(k_i)$ shows the probability of k_i and i represents the intensity of the probability. Mathematically local entropy is represented as:

$$H_{K,T_B}(S) = \sum_{i=1}^{K} \frac{H(S_i)}{K} \tag{12}$$

The calculated entropy values for the proposed algorithm are more significant and close to the ideal entropy (see Table 15).

6.3. Homogeneity

Homogeneity analysis in image processing is a technique that assesses how closely a gray-level co-occurrence matrix (GLCM) conforms to its diagonal. The GLCM is a helpful tool for feature and texture extraction because it measures the frequency of adjacent horizontal pixels with gray-level values of i and j. The resulting homogeneity value ranges from 0 to 1, its lower values reflect better performance.

Mathematically, the homogeneity is calculated as fellow: (see Table 16)

$$Homogeneity = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{13}$$

The aforementioned equation determines the value of a given GLCM element, where $i$and j are two adjacent gray-level values. Table 16 shows the results of the homogeneity of different images and the proposed algorithm give low values of homogeneity.

6.4. Contrast

Contrast analysis is a technique for comparing the variations between a pixel and its neighboring pixels of an entire image. The lightest and darkest areas of an image are referred to this technique. The difference between the light and dark portions is significant in high contrast images. if an image has high contrast, it is considered more secure because it suggests a higher level of randomness. It is represented as:

Table 17
Contrast Results.

Image	Color and Size	Contrast
Airport	1024 × 1024	0.097598
Baboon	512 × 512	0.038494
Chemical Plant	256 × 256	0.084479
Earth Space	512 × 512	0.038193
Tank	512 × 512	0.097193

Table 18
Energy Results.

Image	Color and Size	Energy
Airport	1024 × 1024	2.57488231
Baboon	512 × 512	6.3908717
Chemical Plant	256 × 256	1.5926979
Earth Space	512 × 512	6.3966082
Tank	512 × 512	6.3096318

$$Contrast = \sum_{i,j} |i-j|^2 p(i,j) \tag{14}$$

Table 17 shows the results of the contrast of different images and the proposed algorithm give high contrast levels in encrypted images.

6.5. Energy

Energy, also referred to as uniformity, is a metric that assesses the total squared values in an image's Grey-Level Co-occurrence Matrix (GLCM). A cipher image with a lower energy value is preferred because it is more secure. The energy metric typically has a range of 0 to 1. The mathematical formula for calculating energy is as follows:

$$Energy = \sum_{i,j} p(i,j)^2 \tag{15}$$

Where i and j represent two levels of brightness that appear side-by-side horizontally in an image. Table 18 shows the results of the energy of different images by the proposed algorithm.

6.6. Histogram analysis

A fundamental method for processing images, histogram analysis involves analyzing and interpreting the distribution of pixels in an image. A histogram is a useful tool for analyzing an image's overall brightness, contrast, and tonal distribution because it plots the frequency of pixel intensities against the corresponding intensity values. Common methods derived from histogram analysis include thresholding, contrast stretching, and histogram equalization, which allow for the enhancement, normalization, and segmentation of images. Additionally, histogram analysis is essential for many applications, including image recognition, object detection, and medical imaging.

The following Fig. 5 shows the histogram distribution of the original images:

The following Fig. 6 shows the histogram distribution of the encrypted images:

Based on the analysis, the histogram analysis provides valuable insights into the frequency distribution of images and highlights the presence of distinct peaks, suggesting the presence of multiple groups or subpopulations within the image.

7. Discussion and comparative analysis

In our work, various performance metrics are used to evaluate the encryption security provided by the proposed algorithm for an input image. 1-D and 2-D chaotic maps are mathematical models that exhibit

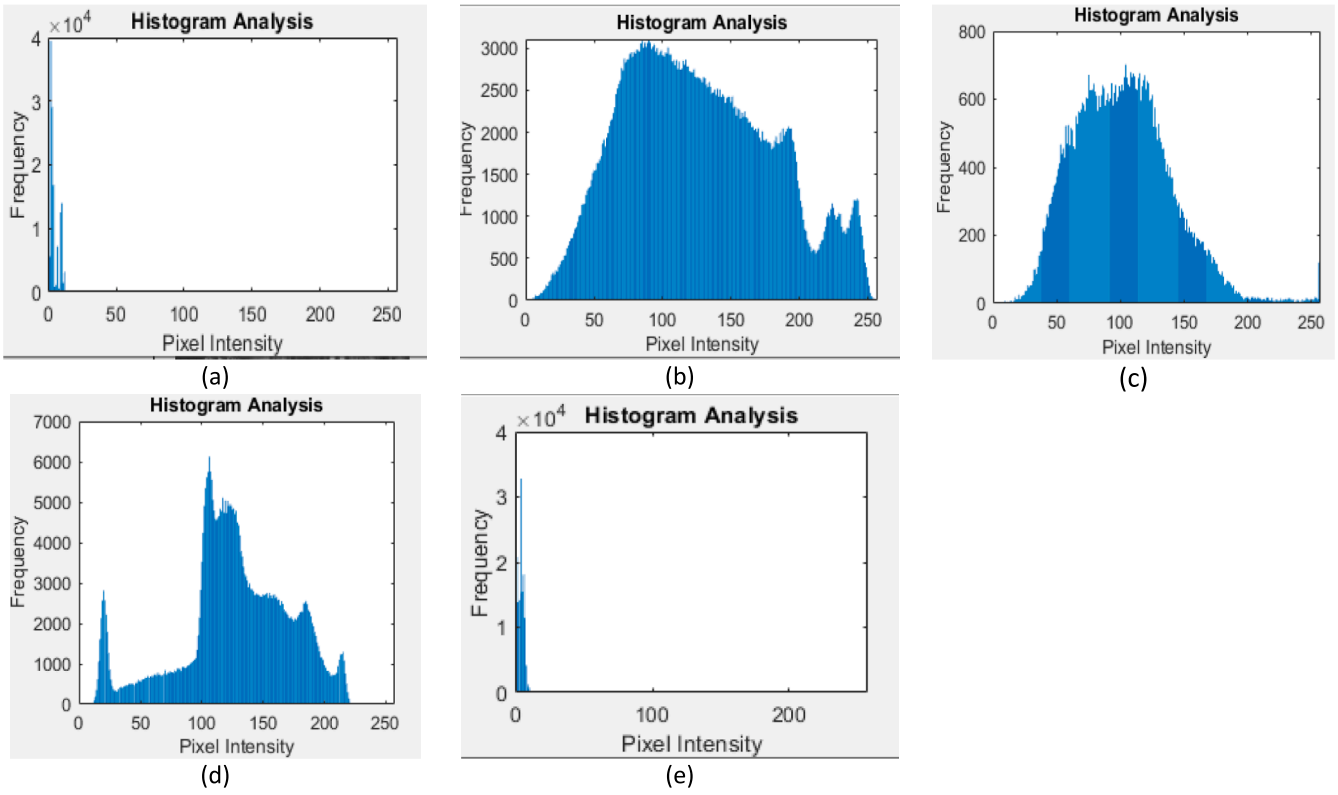

Fig. 5. Histogram of plain images.

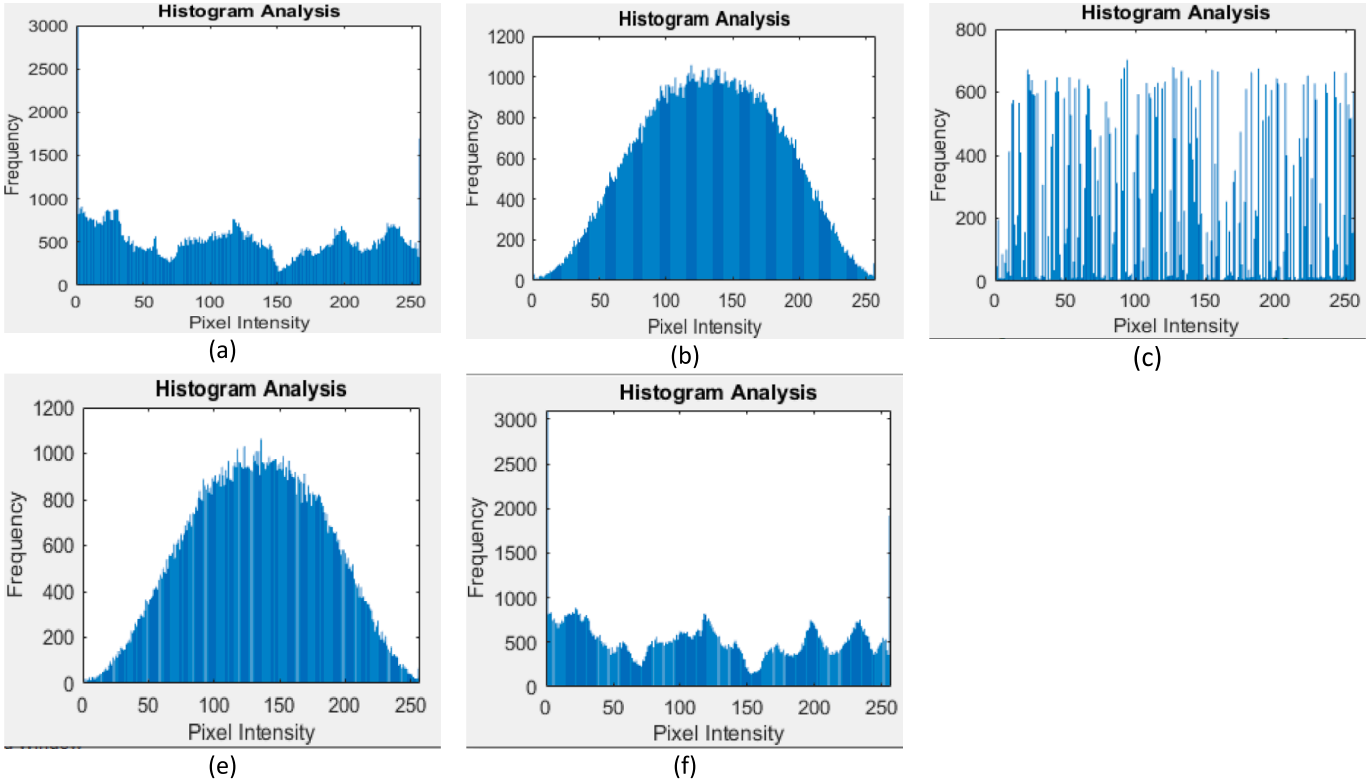

Fig. 6. Histogram Analysis of Encrypted Images.

chaotic behavior. In comparison to 2-D chaotic maps, 1-D chaotic maps are typically easier to compute and analyze [42]. As a result, several attacks can be used against the low-dimensional chaos-based image encryption scheme [43]. The proposed encryption scheme uses 2-D chaotic systems. A variety of test cases are run on a set of color and grayscale images and in our case entropy value approaches to 8, which

Table 19
Comparative Analysis of Proposed Image Encryption Algorithm.

Sr. No.	Encryption Schemes Presented in Literature	Proposed Image Encryption Algorithm
1	Most of the techniques presented in the literature are specific to certain types of images, with some designed for greyscale images and others for RGB images.	The proposed algorithm offers greater practical flexibility by supporting a wide variety of image formats and color representations, including greyscale and RGB images.
2	The outdated encryption methods used by the old algorithms may be vulnerable to attacks and have security flaws.	The proposed algorithm includes sophisticated cryptographic mechanisms that guarantee a higher level of security and resistance to various encryption-breaking techniques.
3	The outdated algorithm might not be able to keep up with new demands for image encryption or changing security threats.	The proposed algorithm is intended to be adaptable, enabling simple updates and improvements to counter new vulnerabilities and address changing encryption requirements.
4	The outdated algorithm might not be sufficiently resilient, making it vulnerable to widely used attacks like statistical analysis or brute force.	The novel algorithm incorporates strong encryption methods that increase its resistance to known attacks and offer improved security for the encrypted images.
5	Most researchers only use a few security analysis measures.	Different metric measures such as Entropy, Contrast, Correlation, homogeneity, and energy are taken into account for results and analysis.

Table 20
Notation Table.

Notation	Notation Name	Description
S-box	Substitution Box	An S-box is a core component of block ciphers.
NL	Nonlinearity	Used to make a complex relationship between input and output.
SAC	Strict Avalanche Criterion	Used to measure the sensitivity of an algorithm to small changes in its input.
BIC	Bit Independence Criterion	When only one input bit is changed, two output bits should change regardless of each other.
DP	Differential Probability	Used to evaluate the security against differential cryptanalysis.
LP	Linear Probability	Used to evaluate the security against linear cryptanalysis.
ν	Nu	Parameter in chaotic map.
μ	Mu	Variable in mathematical equations.
τ	Tau	Variable or parameter in mathematical equations.
ϵ	Epsilon	Parameter to represent the behavior of functions.

indicates presented encryption algorithm is robust and strong.

Similarly, the proposed algorithm exhibits low correlation values between the plaintext and cipher images. This implies a high level of security because it suggests there is little similarity or predictability between the original image and the encrypted image.

In the same way, the analysis section also presents robust findings with satisfactory energy and homogeneity results. The energy analysis shows that computational resources are used effectively during the encryption process, resulting in optimal performance with minimal waste. Additionally, the achieved homogeneity in the encrypted images exhibits a desired degree of uniformity in the distribution of pixel values. This consistency points to a strong encryption algorithm that successfully scrambles the image data while maintaining a reliable and aesthetically pleasing output.

Additionally, the proposed encryption scheme places less computational strain on the system than the high-dimensional chaotic sequence. Table 19 displays the results obtained from the experimental measurements of several standard images using various existing methods, including the model we have proposed. The proposed encryption method is notable for its ability to function well on both RGB and greyscale images (see Table 20).

8. Conclusion

In this study, we propose three chaotic S-boxes based on the Zaslavsky map, Henon map, and Baker's map, which have nonlinearities of 113.75, 114.25, and 112, respectively. We thoroughly examine these S-boxes using cryptographic tests, including nonlinearity, strict avalanche criterion, bit independence criterion, linear approximation probability, differential approximation probability, and bijectivity. The results of these tests are satisfactory and meet our expectations. Additionally, we propose an image encryption algorithm based on our novel S-boxes. To assess the strength and performance of our proposed scheme, we perform statistical tests, including entropy, correlation, homogeneity, energy, and contrast. The experimental results demonstrate that our scheme has good robustness against differential attacks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Moreover, this article is a part of the PhD thesis "Design and Optimization of Nonlinear Component of Block Cipher: Applications to Multimedia Security" submitted to NUML, Pakistan.

References

[1] Chai X, Fu X, Gan Z, Lu Y, Chen Y. A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Process 2019;155:44–62. https://doi.org/10.1016/j.sigpro.2018.09.029.

[2] Chai X, Zhi X, Gan Z, Zhang Y, Chen Y, Fu J. Combining improved genetic algorithm and matrix semi-tensor product (STP) in color image encryption. Signal Process 2021;183. https://doi.org/10.1016/j.sigpro.2021.108041.

[3] Wang L, Cao Y, Jahanshahi H, Wang Z, Mou J. Color image encryption algorithm based on Double layer Josephus scramble and laser chaotic system. Optik (Stuttg) 2023;275. https://doi.org/10.1016/j.ijleo.2023.170590.

[4] Alexan W, Alexan N, Gabr M. Multiple-Layer Image Encryption Utilizing Fractional-Order Chen Hyperchaotic Map and Cryptographically Secure PRNGs. Fractal Fract 2023;7. https://doi.org/10.3390/fractalfract7040287.

[5] Waheed A, Subhan F, Suud MM, Alam M, Ahmad S. An analytical review of current S-box design methodologies, performance evaluation criteria, and major challenges. Multimed Tools Appl 2023. https://doi.org/10.1007/s11042-023-14910-3.

[6] Guan ZH, Huang F, Guan W. Chaos-based image encryption algorithm. Phys Lett Sect A Gen At Solid State Phys 2005;346:153–7. https://doi.org/10.1016/j.physleta.2005.08.006.

[7] Alexan W, Elbeltagy M, Aboshousha A. RGB Image Encryption through Cellular Automata, S-Box and the Lorenz System. Symmetry (Basel) 2022;14. https://doi.org/10.3390/sym14030443.

[8] Bian J, Tao Y, Wang Z, Dong Y, Li Z, Zhao X, et al. A true random number generator based on double threshold-switching memristors for image encryption. Appl Phys Lett 2023;122. https://doi.org/10.1063/5.0145875.

[9] Liang Q, Zhu C. A new one-dimensional chaotic map for image encryption scheme based on random DNA coding. Opt Laser Technol 2023;160. https://doi.org/10.1016/j.optlastec.2022.109033.

[10] Zhao W, Chang Z, Ma C, Shen Z. A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks. Entropy 2023;25. https://doi.org/10.3390/e25010166.

[11] Das D, Pradhan C. Image Encryption Based on Cyclic Chaos, PRNG and Arnold's Cat Map. Lect Notes Networks Syst 2023;572:281–91. https://doi.org/10.1007/978-981-19-7615-5_25.

[12] Alexan W, Chen YL, Por LY, Gabr M. Hyperchaotic Maps and the Single Neuron Model: A Novel Framework for Chaos-Based Image Encryption. Symmetry (Basel) 2023;15. https://doi.org/10.3390/sym15051081.

[13] Waheed A, Subhan F, Suud MM, Malik YH, Al E. Construction of nonlinear component of block cipher using coset graph. AIMS Math 2023;8:21644–67. https://doi.org/10.3934/math.20231104.

[14] Wang J, Yang Y, Wang T, Simon Sherratt R, Zhang J. Big data service architecture: A survey. J Internet Technol 2020;21:393–405. https://doi.org/10.3966/160792642020032102008.

[15] Zhang J, Zhong S, Wang T, Chao HC, Wang J. Blockchain-based Systems and Applications: A survey. J Internet Technol 2020;21:1–14. https://doi.org/10.3966/160792642020012101001.

[16] Bu HH, Kim NC, Park KW, Kim SH. Content-based image retrieval using combined texture and color features based on multi-resolution multi-direction filtering and

color autocorrelogram. J Ambient Intell Humaniz Comput 2019. https://doi.org/10.1007/s12652-019-01466-0.
[17] Liu C, Li K, Li K, Buyya R. A New Service Mechanism for Profit Optimizations of a Cloud Provider and Its Users. IEEE Trans Cloud Comput 2021;9:14–26. https://doi.org/10.1109/TCC.2017.2701793.
[18] Lu L, Ren X, Yeh KH, Tan Z, Chanussot J. Exploring coupled images fusion based on joint tensor decomposition. Human-Centric Comput. Inf Sci 2020;10. https://doi.org/10.1186/s13673-020-00215-z.
[19] Hamza R, Titouna F. A novel sensitive image encryption algorithm based on the Zaslavsky chaotic map. Inf Secur J 2016;25:162–79. https://doi.org/10.1080/19393555.2016.1212954.
[20] Chaos and Cryptography. Baker's map || Chaotic map || Generate chaotic keys || Generate prime numbers [Video] 2021.
[21] Chaos and Cryptography. PRNG with Zaslavasky Map || Pseudo Random Number Generator [Video] 2021.
[22] Chaos and Cryptography. Generation to chaotic values of Henon Map || 2D chaotic map [Video] 2021.
[23] Abduljabbar ZA, Abduljaleel IQ, Ma J, Sibahee MAA, Nyangaresi VO, Honi DG, et al. Provably Secure and Fast Color Image Encryption Algorithm Based on S-Boxes and Hyperchaotic Map. IEEE Access 2022;10:26257–70. https://doi.org/10.1109/ACCESS.2022.3151174.
[24] Guisande N, Di Nunzio MP, Martinez N, Rosso OA, Montani F. Chaotic dynamics of the Hénon map and neuronal input-output: A comparison with neurophysiological data. Chaos 2023;33. https://doi.org/10.1063/5.0142773.
[25] Alhumyani H. Dual Image Cryptosystem Using Henon Map and Discrete Fourier Transform. Intell Autom Soft Comput 2023;36:2933–45. https://doi.org/10.32604/iasc.2023.034689.
[26] Abd-El-Atty B. Efficient S-box construction based on quantum-inspired quantum walks with PSO algorithm and its application to image cryptosystem. Complex Intell Syst 2023. https://doi.org/10.1007/s40747-023-00988-7.
[27] Zhu D, Tong X, Zhang M, Wang Z. A new s-box generation method and advanced design based on combined chaotic system. Symmetry (Basel) 2020;12:1–17. https://doi.org/10.3390/sym12122087.
[28] Zahid AH, Iliyasu AM, Ahmad M, Shaban MMU, Arshad MJ, Alhadawi HS, et al. A Novel Construction of Dynamic S-Box with High Nonlinearity Using Heuristic Evolution. IEEE Access 2021;9:67797–812. https://doi.org/10.1109/ACCESS.2021.3077194.
[29] Alghafis A, Munir N, Khan M. An encryption scheme based on chaotic Rabinovich-Fabrikant system and S8 confusion component. Multimed Tools Appl 2021;80:7967–85. https://doi.org/10.1007/s11042-020-10142-x.
[30] Jiang Z, Ding Q. Construction of an s-box based on chaotic and bent functions. Symmetry (Basel) 2021;13. https://doi.org/10.3390/sym13040671.
[31] Dimitrov MM. On the Design of Chaos-Based S-Boxes. IEEE Access 2020;8:117173–81. https://doi.org/10.1109/ACCESS.2020.3004526.
[32] Shahzad I, Mushtaq Q, Razaq A. Construction of New S-Box Using Action of Quotient of the Modular Group for Multimedia Security. Secur Commun. Networks 2019;2019. https://doi.org/10.1155/2019/2847801.
[33] Hua Z, Li J, Chen Y, Yi S. Design and application of an S-box using complete Latin square. Nonlinear Dyn 2021;104:807–25. https://doi.org/10.1007/s11071-021-06308-3.
[34] Javeed A, Shah T, Attaullah. Design of an S-box using Rabinovich-Fabrikant system of differential equations perceiving third order nonlinearity. Multimed Tools Appl 2020;79:6649–60. https://doi.org/10.1007/s11042-019-08393-4.
[35] Ali TS, Ali R. A novel color image encryption scheme based on a new dynamic compound chaotic map and S-box. Multimed Tools Appl 2022;81:20585–609. https://doi.org/10.1007/s11042-022-12268-6.
[36] Hematpour N, Ahadpour S, Behnia S. Presence of dynamics of quantum dots in the digital signature using DNA alphabet and chaotic S-box. Multimed Tools Appl 2021;80:10509–31. https://doi.org/10.1007/s11042-020-10059-5.
[37] Nizam Chew LC, Ismail ES. S-box construction based on linear fractional transformation and permutation function. Symmetry (Basel) 2020;12. https://doi.org/10.3390/SYM12050826.
[38] Artuğer F, Özkaynak F. SBOX-CGA: substitution box generator based on chaos and genetic algorithm. Neural Comput Appl 2022. https://doi.org/10.1007/s00521-022-07589-4.
[39] Panchami V, Mathews MM. A Substitution box for Lightweight Ciphers to Secure Internet of Things. J King Saud Univ - Comput Inf Sci 2023. https://doi.org/10.1016/j.jksuci.2023.03.004.
[40] Cheng R, Zhou Y, Miao X, Hu J. Analysis of a New Improved AES S-Box. Structure 2023:109–25. https://doi.org/10.1007/978-3-031-30623-5_8.
[41] Webster AF, Tavares SE. On the Design of S-Boxes. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 1986;218 LNCS:523–34. https://doi.org/10.1007/3-540-39799-X_41.
[42] Zhou S, Wei Y, Zhang Y, Teng L. Novel Chaotic Image Cryptosystem Using Dynamic DNA Coding 2022.
[43] Solak E, Çokal C, Yildiz OT, Biyikoğlu T. Cryptanalysis of fridrich's chaotic image encryption. Int J Bifurc Chaos 2010;20:1405–13. https://doi.org/10.1142/S0218127410026563.