International Journal of Computational and Electronic Aspects in Engineering
Volume 5, Issue 4, December 2024, pp 194 - 202
https://doi.org/10.26706/ijceae.5.4..20241107

Use of Chaos in Key Schedules for A Symmetrical Encryption Algorithm with Out Data Loss

Serri Ismael Hamad

College of Education for Pure Sciences, University of Thi-Qar, Nasiriyah, Iraq

*Correspondence: serriismael@utq.edu.iq

Abstract: This research presents the construction of a symmetric encryption algorithm for images of 512 × 512 pixels, in color and grayscale without loss of information and without compression, in BMP format. This algorithm consists of 15 rounds with S-box swapping, permutations, X-OR operations, and image size keys. The logistic map operation was used for the generation of S-boxes and to build a schedule of keys of the size of the image, the equations of E. Lorenz, transcendental numbers and the elliptic curve were used. To measure the strength of the encrypted result, the entropy of the information and the goodness test $\chi\wedge 2$ were used, obtaining favorable results and close to similar algorithms.

Keywords: Image Encryption; Chaos; Elliptic Curve, SPN.

Article – Peer Reviewed
Received: 15 Sept 2024
Accepted: 29 Nov 2024
Published: 30 Dec 2024

Cite this article: Serri Ismael Hamad, "Use of Chaos in Key Schedules for A Symmetrical Encryption Algorithm with Out Data Loss", *International Journal of Computational and Electronic Aspects in Engineering*, RAME Publishers, vol. 5, issue 4, pp. 194-202, 2024.
https://doi.org/10.26706/ijceae.5.4.20241107

1. Introduction

This work presents an information protection algorithm through the use of data encryption, especially 512 × 512 pixel images without loss and without compression for the following reasons: NOM-151 determines how digital documents should be processed and preserved [1] since there are images with sensitive content such as in the areas of medicine, military, finance, etc. Although research highlights the use and management of the JPG format due to its common use, due to its format characteristics there is data loss [2], adding that there are algorithms that compress data [3]. Under the first reason, this development focuses on encrypting images in BMP format and without data compression, this will allow to protect the mentioned objects, as mentioned in the following developments: [4–6]. This algorithm is symmetric of the Substitution Permutation Network (SPN) type which consists of 15 rounds, substitution boxes (S-box) and a dynamic key schedule and will be called SecCaos-Image for the following reasons:

1- Although there is a mathematical basis for the SPN proposal, it can be adapted with various innovations such as the use of certain tools in the development and construction of its S boxes, key schedule and even permutations.

2- SecCaos-Image uses the logistic map equation for the construction of its S-boxes and part of SPN for its constitution. In the case of [7], SPN is proposed and the logistic map equation is used for the construction of the boxes and the key schedule. For the S-box, there are some other cases where the chaotic map is applied [8].

3- For this development, the E. Lorenz equations are used in this improvement of the key schedule of the image size and in conjunction with transcendental numbers such as e and the elliptic curve. In some other important works these tools were applied in a close manner or to develop some other module of their algorithm [9–11].

4- To verify the resistance of the encrypted products, information entropy analysis was applied to the keys and images, image correlation, chi-square $\chi 2$, as well as an analysis of the S-boxes.

It is proposed that SecCaos-Image be symmetric based on rounds, with the ability to bet on uncertainty, since, as is known, the asymmetric RSA algorithm has the vulnerability that through the calculation of a prime number by means of the short algorithm applied in a post-quantum analysis, it is possible to compute efficiently [12], otherwise, until now, symmetric encryption algorithms still show resistance to this type of calculations.

2. General Considerations

2.1 SPN network

A SPN algorithm is a network that consists of dividing a clear text message into blocks of bits of equal size (2^n); in this case it will consist of 15 rounds to carry out the substitutions and permutations in each block. In each round r, a key k from the set of keys K and an expansion function E are used. The list of operations of $E_k(K)$ will be equal to K^1, … $^{Knr+1}$ which will result in a public algorithm [13].

For this case, two permutations are proposed: a different S-box substitution box for each encryption process called π_s; π_p, which permutes the bit positions of each block, where l and m are positive integers, as shown in Eq. (1).

$$\pi_s: \{0,1\}^l \rightarrow \{0,1\}^l \qquad \pi_p: \{1,\ldots,lm\} \rightarrow \{1,\ldots,lm\} \tag{1}$$

Thus, the plaintext of Eq. (2), has length lm, and x is interpreted as a concatenation of m bit strings, each string containing l bits.

$$x = (x_1, \ldots, x_{lm}) \tag{2}$$

Eq. (3), represents $x(i)$:

$$x(i)=x(i-1)l+1,\ldots,xil) \tag{3}$$

2.2. Chaos

Chaos is applied in this algorithm to develop chaos in the S-boxes and in the keys; in the first case, the logistic map equation defined in Eq. (4) is used [14]:

$$x_{n+1}= r \times x_n(1-x_n) \tag{4}$$

Where r it has the value of 3.8817182818…, and it is applied with a length of more than 300 decimal places. The range of the variable x is $0 < x < 1$. This satisfies that x_n is deterministic, in turn, any change to or x_0. In short, x_n cannot be predicted without prior calculations.On the other hand, E. Lorenz differential equations are applied to build the set of keys of the key size in pixels. This system of equations is shown in Eq. (5) [14]:

$$\begin{aligned} \frac{dy}{dt} &= \sigma(y-x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dy}{dt} &= -bz + xy \end{aligned} \tag{5}$$

The values of the parameters $\sigma, r\ y\ b$, determine the chaotic behavior. SecCaos-Image uses the following values shown in Eq (6):

$$\begin{aligned} \frac{dy}{dt} &= -10(y-x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dy}{dt} &= xy - \frac{8z}{3} \end{aligned} \tag{6}$$

This system of equations was solved by taking its angles in radians to apply the number e. To solve Eq (6), e is raised to a natural power; finally, the points to be generated from the elliptic curve are applied to obtain a result to the final equation of this system.

2.3 Entropy

Information entropy is a measure applied to measure the quality of encryption and is calculated according to Eq. (7) [14–16]:

$$H(x)=-\sum_{xsX}P(x)log_2P(x) \quad \textbf{(7)}$$

The image is treated as a matrix, where the pixels are extracted in RGB color format (red, green and blue); each color will be a Byte, ∴ , there are 2^n color values. Due to the nature of the digital image, n=8 with 256 values in total. If the probability of occurrence of all events is 1, the maximum entropy value is 8. In practice, an entropy close to 8 is sought if the distribution of bits is uniform, however, it may mean that they are not random, so it is proposed to support it with more measurement instruments.

2.4 Correlation

Statistical measurement that is carried out between two random bits. It is used to measure the relationship index between these variables. This analysis is carried out in 3 directions: horizontal, vertical and diagonal. A pixel (RGB) is taken randomly between the values of 0 and 255. The representation of these is as follows: Y_r for red; Y_v for green; Y_a for blue. The pixel adjacent to the newly selected one is obtained and compared in the three directions mentioned. These values are represented as: Z_r, Z_v and Z_a for red,green and blue respectively. Eq. (8) [15] represents the correlation:

$$r_{k;Y_r,Z_{r,}}=\frac{\frac{1}{N}(\sum_{i=1}^{N}(Y_{i,r}-\bar{Y}r)(Z_{i,r}-\bar{Z}r))}{\sqrt{\frac{1}{N}(\sum_{i=1}^{N}(Y_{i,r}-\bar{Y}r)^2)(\frac{1}{N}\sum_{i=1}^{N}(Z_{i,r}-\bar{Z}y)^2}} \quad \textbf{(8)}$$

2.5 Goodness-of-fit test

The chi-square goodness-of-fit test χ^2 is a study that determines whether two or more elements are normal. This test shows whether the values of the color intensities (RGB) have a uniform distribution. If so, the distribution is random. In this case, the χ^2 < 308 must be fulfilled for the encrypted product to be random. Eq.(9) shows the χ^2 where O_i is the observed value; $expi$ is the expected quantity [15], [16].

$$\chi^2=\sum_{i=1}^{i=k}(\frac{O_i-exp_i}{exp_i})^2 \quad \textbf{(9)}$$

3. Encryption Elements

3.1. Permutation algorithm

The use of the permutation generation algorithm used in Ref [17] is used. It starts from a non-negative integer $m \geq 2$ with the following sets N_m ={$n\varepsilon N$}|$0 \leq n \leq m!-1$ and Π_m={π}. It is noted that π is a permutation of the array 0,1,…,m-1. As explained in the previous reference, the Euclidean division algorithm is applied, given that $n\ \varepsilon\ N$, to obtain Eq.(10):

$$n= C_0(m-1)!+C_1(m-2)!+\cdots+C_{m-2}(1)!+C_{m-1}(0)! \quad \textbf{(10)}$$

Based on m, $(m-1)!$, $(m-2)!$, … ,1! ,0! are fixed.

An example of the permutation when m=8 is shown, N_8 = {$n\ \varepsilon\ N$}| $0 \leq n \leq 8!-1$, and the permutation Π_m = {$\pi|\pi\ es\ una\ permutacin\ del\ arreglo$ 0,1, … ,7}. n will be 21699, then Eq. (11) is expressed

$$21699=4(7!)+2(6!)+0(5!)+4(4!)+0(3!)+1(2!)+1(1!)+0(0!) \quad \textbf{(11)}$$

It is obtained that C_0=4;C_1=2;C_2=0;C_3=4;C_4=0;C_5=1;C_6=1andC_7=0, so n=42075163. This is represented in Table 1.

Table 1. Permutation table

$C_0 = 4$	$C_1 = 2$	$C_2 = 0$	$C_3 = 4$	$C_4 = 0$	$C_5 = 1$	$C_6 = 1$	$C_7 = 0$
0	0	0	5	5	3	3	3
1	1	1	1	1	1	6	
2	2	6	6	6	6		
3	3	3	3	3			
4	7	7	7				
5	5	5					
6	6						
7							

3.2. Building an S-box

We start from Eq. (4) with values for r= 3.88171…, with a length of 313 digits. The procedure is as follows:

- x_0 is the random starting value with values between $0 < x_0 < 1$.
- We iterate until n=10000, where n < 1. This number gives a set of decimals that do not follow a pattern. For each iteration, one thousand numbers are taken in hexadecimal format.
- We calculate the constants of Eq. (10), where $C_i = b_i$, where b_i is the value associated with a block (byte) after the decimal point.
- We apply the following algorithm:

Algorithm 1. Generating Permutations

Input : Image size h
Apply X[0]= 0.. . X[h−1]= h−1
For i←0 **to** h−1 do
if i=h−1 **then**
Y[i]=X[C_i]
end
else if C_i=h−1− i **then**
Y[i]=X[C_i]
end
else Y[i] = X[C_i]
X[C_i]=X[h−1−i]
end **end**
end
output Y[i]

3.3. Encryption Procedure

Figure (1) shows the encryption process that involves breaking down an image into a matrix, where the pixels in RGB format are taken to form the plain text (PT); with the size of the image, the permutation is built by calculating each constant

C_i randomly using the number π of the decimal point on the right; An operation $\oplus$ is applied with the key k_{i+1}, fulfilling the following premises:

1- K is a 512-bit string associated with the positive integer l. The product $l \times \pi$ is performed, from which a 24-bit block is selected, with the argument that the images used in this experiment do not exceed a resolution of 2^{24}.

2- By calculating the constants of the permutation C; it is established that $C_j = aj \bmod h - j$, where h is the size of the image in pixels.

3- Based on the previous algorithm, π_h is associated

4- From the key schedule K of length h, 14 keys k_i are selected randomly $l \times \pi$, starting from the decimal point on the right; A block of bits of the image size D_0 is taken and an xor operation is performed on the key k_i.

5- The calculated S-box is applied up to round R_{13}.

6- The previous result uses an inverse permutation $\boldsymbol{\pi_h}^{-1}$to the string R.

7- Finally, the substitution is performed one-sidedly to k_{14}.

To illustrate this point, an 8x8 S-box was randomly generated in hexadecimal format, which is shown in Table 2. It is worth mentioning that in each encryption-decryption process, a different S-box is created and integrated, therefore, the characteristics of these will vary and will present different virtues.

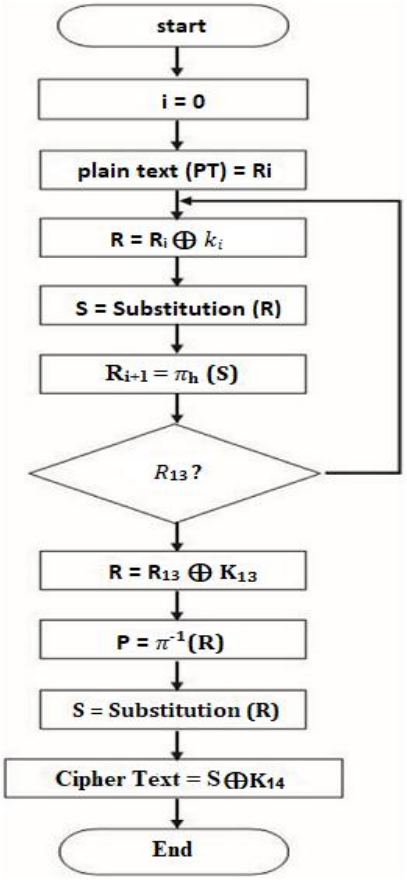


Figure 1. Encryption Process.

Table 2. S-box generated in hexadecimal format.

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

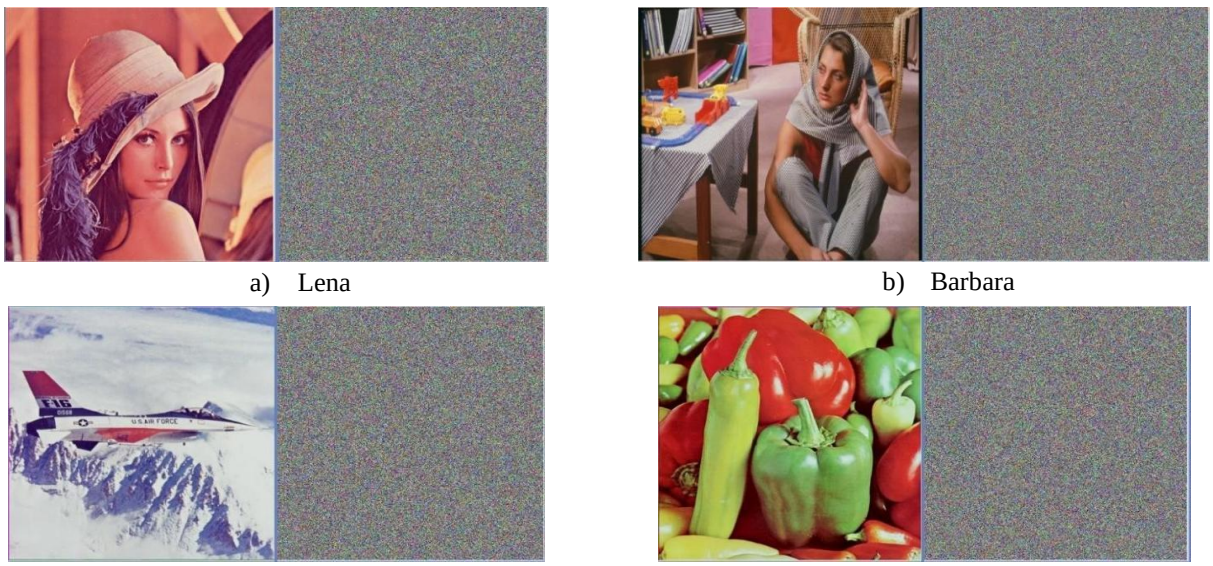
Table 3. S-box characteristics for an encryption-decryption process

Balance	0
Non linearity	94
Absolute indicator	96
Sum of square indicator	259840
Core lation immunity	0
Algebraic degree	7
Transparencyorder	7.812
Propagation characteristic	0
Number of opposite fixed points	1
Composite algebraic immunity	4
Robustness to differential cryptanalysis	0.961
Delta uniformity	10
SNR(DPA)(F)	9.867
Confusion coefficient variance	0.102003

Table 3 shows an S-box generated in an encryption-decryption process. Although there are works where a high non-linearity in the S-boxes is highlighted, such as Ref [18], where it is shown that the strength of some encryption algorithms is centered on the construction of a high non-linearity S-box which on average gives a value of 115.75, SecCaos-Image bets that, in each execution, a different S-box is built and integrated into the process, in this way, an element of uncertainty will be added to the face of a possible attack on the encryption process. In the same sense, generating and building a high non-linearity S-box may require considerable time and resources, as shown in Table 10 of the previous reference.

3.4. Images used

The images proposed for this research are in the public domain and are visualized in Figure (20). They are well-known objects for image processing in BMP format of 512 × 512 pixels based on the literature attached to this work.



a) Lena b) Barbara

c) Jet d) Peppers

Figure 2. Images to be examined.

4. Analysis of Results

4.1. Encrypted Images

Below are the images encrypted with SecCaos-Image. As shown in Figure (3), Figure (4), Figure (5) and Figure (6), the encryption process took place, where at first glance no pattern related to the content of the original image can be distinguished.

Figure 3. Image of encrypted Lena sincifrary

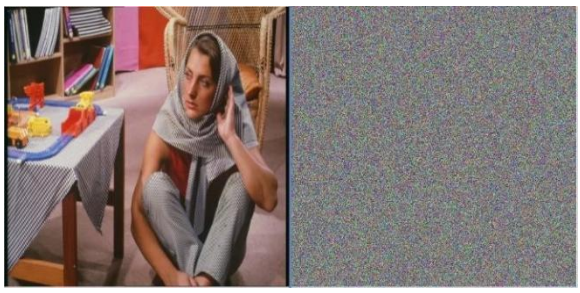


Figure 4. Image of Barbara unencrypted and encrypted

Figure 5. Image of unencrypted and encrypted Jets

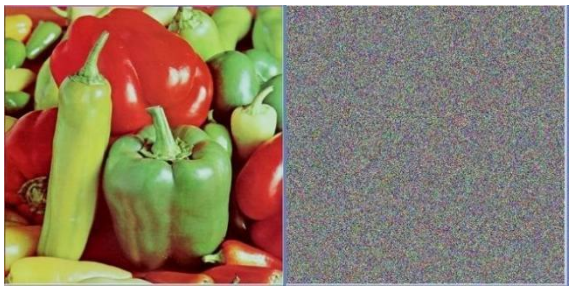


Figure 6. Unencrypted and encrypted Peppers image

4.2. Entropy analysis

As an object of study of this research, an analysis was performed on the set of keys K_h by applying the Entropy of information. Table 4 shows the average entropy of the 14 keys in each of the images. Although most research focuses its studies on the entropy of encrypted products, it is considered important for this development to analyze the entropy of the keys, since they are the strength of a symmetric encryption system.

Table 4. Entropy of the keys

Imagen	Entropy increase
Lena	7.99977
Barbara	7.99985
Jet	7.99977
Peppers	7.99976

Table (5) below shows the entropy of the encrypted objects in order to show a comparison with the literature presented. Although the results presented may be very close, SecCaos-Image has an entropy of 7.999 in most of its results, compared to the rest that mostly have an entropy of 7.99.

Table 5. Entropy of images encrypted with SecCaos-Image

Imagen	SecCaos-Image	Ref [18]	Ref [10]	Ref [9]
Lena	7.99933	7.9971	7.9974	7.9997
Barbara	7.99954	7.9967	-	-
Jet	7.99926	7.9973	-	7.9980
Peppers	7.99921	7.9975	7.9969	7.9971

4.3. Correlation Analysis

Table (6) shows the correlation results of the encrypted images in Figure (3), Figure(4), Figure 5), and Figure (6). In these cases, the correlation between the points is shown to be close to zero.

Table 6. Correlation of images encrypted with SecCaos-Image

Imagen	SecCaos-Image	Ref [18]	Ref [10]	Ref [9]
Lena	-0.0004	0.0203	-0.0012	-0.0034
Barbara	0.0091	-0.0138	-	-
Jet	0.0082	-0.0058	-	0.0084
Peppers	-0.0062	-0.0045	-0.0013	-0.0039

4.4. Analysis of the goodness test (χ^2)

Table (7) presents the chi-square values for the encrypted images, all of which are acceptable and confirm the randomness. In the case of [9] the evaluation presented was carried out in a different way, so its values do not represent a value to compare in this case.

Table 7. Chi-square analysis of images encrypted with SecCaos

Imagen	SecCaos-Image	Ref [18]	Ref [10]	Ref [9]
Lena	246.46	266.16	234.15	-
Barbara	256.76	296.71	-	-
Jet	266.4	246.66	-	-
Peppers	285.4	230.5	236.10	

5. Conclusions

SecCaos-Image is a symmetric image encryption algorithm, which bases its strength on its key schedule, which were posed under the set of differential equations of E. Lorenz, the transcendental number e and random points of the elliptic curve, which can even be chosen by some other method to give an answer; there were 2^{512} keys of object length. On the other hand, a method described and used in other investigations was applied to develop a different S-box for each execution, as well as a variable permutation. It was possible to confirm a robust application and algorithm that can resist brute force and differential attacks, since it has a very high computational cost derived from the set of keys, the S-box, the variable permutation and the number of rounds. The results of the tests were acceptable and in most cases, some attributes of this development are superior to those compared. The development worked for images in BMP format, and at the moment it is not compatible with any type of data compression. Finally, the application was developed in Java 8 with FX interface, with an image processing time of less than 0.4s for each reading, display and encryption-decryption of each figure.

References

[1] M. A. Lone and S. Qureshi, "RGB image encryption based on symmetric keys using Arnold transform, 3D chaotic map and affine hill cipher," *Optik*, vol. 260, p. 168880, 2022.

[2] M. Gupta, V. P. Singh, K. K. Gupta, and P. K. Shukla, "An efficient image encryption technique based on two-level security for internet of things," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 5091–5111, 2023.

[3] B. Zhang, D. Xiao, H. Huang, and J. Liang, "Compressing cipher images by using semi-tensor product compressed sensing and pre-mapping," in *2022 Data Compression Conference (DCC)*, 2022, pp. 123–132.

[4] F. H. M. Al-Kadei, "Two-level hiding an encrypted image," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 961–969, 2020.

[5] M. A. Budiman and M. Y. Saputra, "A hybrid cryptosystem using Vigenère cipher and Rabin-p algorithm in securing BMP files," *Data Science: Journal of Computing and Applied Informatics*, vol. 4, no. 2, pp. 89–99, 2020.

[6] G. Manikandan, R. Bala, E. Preethivi, K. Sekar, R. Manikandan, and J. Prassanna, "An approach with steganography and scrambling mechanism for hiding image over images," *International Journal on Emerging Technologies*, vol. 10, no. 1, pp. 64–67, 2019.

[7] M. F. Khan, A. Ahmed, K. Saleem, and T. Shah, "A novel design of cryptographic SP-network based on gold sequences and chaotic logistic tent system," *IEEE Access*, vol. 7, pp. 84980–84991, 2019.

[8] A. S. Alanazi, N. Munir, M. Khan, and I. Hussain, "A novel design of audio signals encryption with substitution

permutation network based on the Genesio-Tesi chaotic system," *Multimedia Tools and Applications*, vol. 82, no. 17, pp. 26577–26593, 2023.

[9] B. Ahuja, R. Doriya, S. Salunke, M. F. Hashmi, A. Gupta, and N. D. Bokde, "HDIEA: high dimensional color image encryption architecture using five-dimensional Gauss-logistic and Lorenz system," *Connection Science*, vol. 35, no. 1, p. 2175792, 2023.

[10] M. Ramzan, T. Shah, M. M. Hazzazi, A. Aljaedi, and A. R. Alharbi, "Construction of s-boxes using different maps over elliptic curves for image encryption," *IEEE Access*, vol. 9, pp. 157106–157123, 2021.

[11] C. Zou, Q. Zhang, X. Wei, and C. Liu, "Image encryption based on improved Lorenz system," *IEEE Access*, vol. 8, pp. 75728–75740, 2020.

[12] S. Rass and P. Schartner, "Authentic quantum nonces," in *Quantum Random Number Generation: Theory and Practice*, Springer, 2020, pp. 35–44.

[13] R. Civino, C. Blondeau, and M. Sala, "Differential attacks: using alternative operations," *Designs, Codes and Cryptography*, vol. 87, pp. 225–247, 2019.

[14] M. A. Murillo-Escobar, M. O. Meranza-Castillón, R. M. López-Gutiérrez, and C. Cruz-Hernández, "A chaotic encryption algorithm for image privacy based on two pseudorandomly enhanced logistic maps," *Multimedia Security Using Chaotic Maps: Principles and Methodologies*, pp. 111–136, 2020.

[15] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "Comparison of entropy calculation methods for ransomware encrypted file identification," *Entropy*, vol. 24, no. 10, p. 1503, 2022.

[16] S. N. Ray and S. Chattopadhyay, "Analyzing surface air temperature and rainfall in univariate framework, quantifying uncertainty through Shannon entropy and prediction through artificial neural network," *Earth Science Informatics*, vol. 14, no. 1, pp. 485–503, 2021.

[17] V. M. S. García, M. D. G. Ramírez, R. F. Carapia, E. Vega-Alvarado, and E. R. Escobar, "A novel method for image encryption based on chaos and transcendental numbers," *IEEE Access*, vol. 7, pp. 163729–163739, 2019.

[18] A. H. Zahid *et al.*, "A novel construction of dynamic S-box with high nonlinearity using heuristic evolution," *IEEE Access*, vol. 9, pp. 67797–67812, 2021.