Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq

Image Encryption Techniques Using Dynamic Approach : An Article Review

Wisam Abed Shukur
Department of Computer, College of Education for Pure Sciences, Ibn Al-Haitham, University of Baghdad, Iraq

Zaid M. Jawad Kubba*
Department of Computer, College of Education for Pure Sciences, Ibn Al-Haitham, University of Baghdad, Iraq

Ahmed Badrulddin
Baghdad University Presidency, Law Affairs Department, University of Baghdad, Baghdad, Iraq.

Diaa Mohammed Uliyan
Department of Information and Computer Science, College of Computer Sciences, University of Ha'il, Ha'il, Saudi Arabia.

***Corresponding Author:** zaidkubba@colaw.uobaghdad.edu.iq

Article history: Received 2 March 2023, Accepted 3 May 2023, Published in October 2023

doi.org/10.30526/36.4.3303

Abstract

In this study, dynamic encryption techniques are explored as an image cipher method to generate S-boxes similar to AES S-boxes with the help of a private key belonging to the user and enable images to be encrypted or decrypted using S-boxes. This study consists of two stages: the dynamic generation of the S-box method and the encryption-decryption method. S-boxes should have a non-linear structure, and for this reason, K/DSA (Knutt Durstenfeld Shuffle Algorithm), which is one of the pseudo-random techniques, is used to generate S-boxes dynamically. The biggest advantage of this approach is the production of the inverted S-box with the S-box. Compared to the methods in the literature, the need to store the S-box is eliminated. Also, the fabrication of the S-box has a very large key space as it depends on the user's key. The encryption-decryption method allows changing pixel positions with the help of dynamically generated S-boxes, images, videos, etc. Thus, the study shows that a new method of S-boxes for dynamic cipher algorithms can be easily generated and applied to image encryption.

Keywords: image encryption; dynamic encryption techniques; DES (Data Encryption Standard).

1. Introduction

Dynamic Encryption is widely used to securely conceal and transmit data over an unsecured network. The algorithms used in encryption form the core component of the encryption system. The cipher system consists of an encryption algorithm, a key, plaintext, and ciphertext [1]. In the last century, many methods, such as DES (Data Encryption Standard) and AES (Advanced Encryption Standard), have been developed to encrypt data [2, 3]. Most cipher algorithms are based on block encryption algorithms. Block cipher algorithms such as AES and DES play a very active role in encrypting text-type data. However, due to the large size of image and video files, using them to encode images is not considered appropriate [4].

Nonlinear systems are mainly designed on dynamic models that are sensitive to initial seeds and present randomness keys, which are the best enhancements for developing cryptosystems [2]. In contrast, issues such as confidentiality and integrity are core to implementing cipher algorithms. However, there are a huge number of cipher algorithms based on symmetric and asymmetric approaches, which involve two dissimilar types of cryptography algorithms based on stream and block cryptosystems.

Block-cipher algorithms occupy a very important place in cryptography today [5]. These algorithms use the same key to encrypt and decrypt data. Block cipher algorithms get stronger using dynamic keys. This approach can add hashing capability to prevent cryptographic algorithms, which is the only nonlinear feature [6]. Therefore, a well-chosen S-box directly affects the coding result. While designing S-boxes, pseudo-random finite-field inversion, finite-field inversion, and heuristic techniques are used [5]. Today, with the widespread use of mobile technologies, ensuring the security of image data is also of crucial importance. This security can be ensured by encrypting the images. Image encryption approaches are grouped into two groups: chaotic and non-chaotic techniques. Dynamic encryption algorithms take parameters from the user for encryption and decryption operations [2].

In addition, the key value used in the encryption process changes at each step of the process. If the same user parameter used for encryption is not used in the decoding process, the original image cannot be obtained [6]. Image encryption techniques have three basic ideas. In contrast, this method can be applied based on the substitution, value conversion, and permutation techniques as they are used together [7]. In image encryption, the current pixel color value is changed using value conversion algorithms. Moreover, in displacement algorithms, the positions of existing pixels are shuffled by moving them to other positions with the help of an algorithm. Thus, the dynamic encryption systems should be explored and examined based on nonlinear equations, some analysis must be calculated to show how the image encryption system is introduced, and the main properties of such systems should be examined.

2. Materials and Methods

In this section, S-boxes are explored and shown how they are generated dynamically in order to encipher images and in what manner the generated S-boxes are used in the encryption or decryption of the image.

2.1 K/DSA Algorithm

Random number permutations, or "shuffles," as they are known in the literature, are frequently used in computer calculations as well as in our daily lives in areas such as card shuffling, numerology, cryptography, and simulation [8]. This version of Shuffle Algorithms was published [9] in 1963 and [10] in 1964. But it is known as the Knuth Shuffle and was published in the second volume of The Art of Computer Programming in 1969 [11].

K/DSA is designed to mix a group of elements by exchanging them. The ordered integers from 0 to (n-1) are written into the matrix X. Then a random number j is chosen such that 0 j ≤ (n-1). Then the elements X [i] and X [j] are swapped. In the next step, a new random j is selected; this time, 1 j ≤ (n-2), X[j], and X[i-1] elements are replaced. The process continues until the last two digits are swapped. Of course, in the following steps, while the items moved to the right side cannot be moved again, the places of the items moved to the left will be changed frequently [12]. Due to the nature of K/DSA, if the initial value is the same for random number generation, the same sequence of X will always be obtained.

2.2 S-Box Production with K/DSA

Using the K/DSA method, which is detailed above, the S-Box that will be used for encryption of the images, is dynamically obtained. Dynamically getting the S-box and not storing it is an important feature of image encryption and decryption [13]. Because producing S-boxes suitable for encryption and decryption is only possible if the correct key is known. The S box consists of 256 different integers. So, since its maximum length will be n_{max} =256, the probability of making the correct S-Box can be computing by the following equation:

$$p = 1/\sum_{i=1}^{n_{max}}(n_{max} - i) \quad (1)$$

The probability of obtaining an inverted S-box in one trial is the same as in Equation 1. Creating an inverted S-box with an S-box significantly reduces the decoding time. The flow chart for obtaining an S-Box is shown in Figure 1.

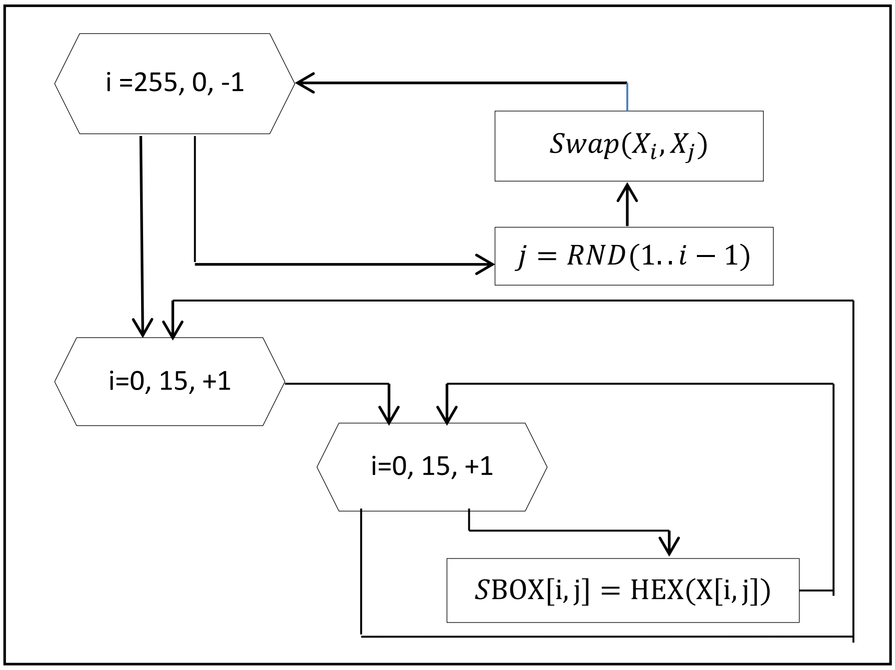
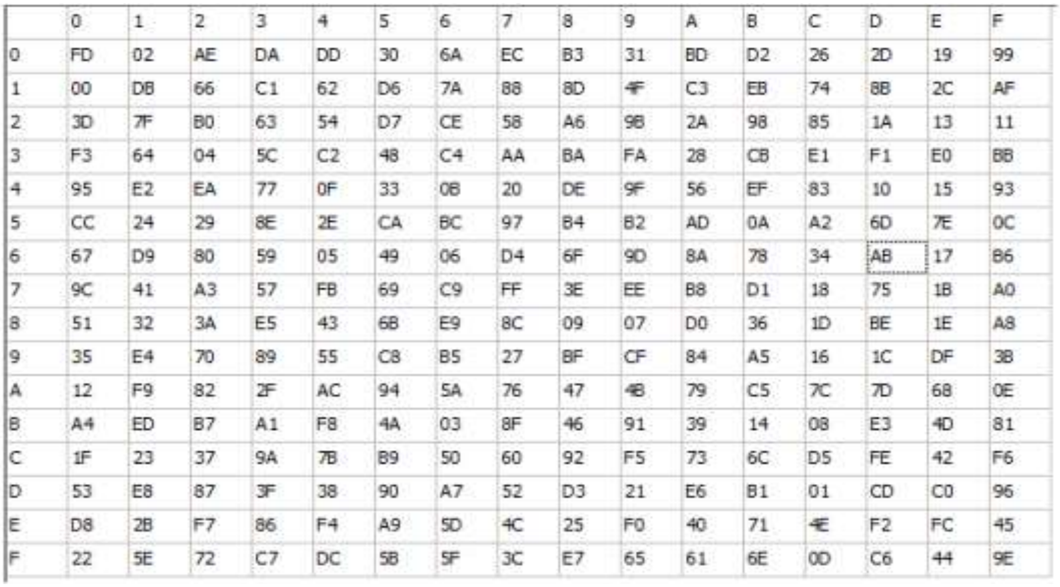


Figure 1. Obtaining an S- Box [13]

S-boxes are 16×16 in size and consist of 256 values. With the help of K/DSA, 256 integers are shuffled according to the initial value set by the user [13, 14]. Starting with the first element of the randomly shuffled array, the hexadecimal equation (16) is taken for each value. The hexadecimal equivalents are placed line by line, starting with the first cell in the 16×16 matrix. One of the S-boxes obtained by K/DSA.is shown in Fig. 2.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FD	02	AE	DA	DD	30	6A	EC	B3	31	BD	D2	26	2D	19	99
1	00	DB	66	C1	62	D6	7A	88	8D	4F	C3	EB	74	8B	2C	AF
2	3D	7F	B0	63	54	D7	CE	58	A6	9B	2A	98	85	1A	13	11
3	F3	64	04	5C	C2	48	C4	AA	BA	FA	28	CB	E1	F1	E0	BB
4	95	E2	EA	77	0F	33	0B	20	DE	9F	56	EF	83	10	15	93
5	CC	24	29	8E	2E	CA	BC	97	B4	B2	AD	0A	A2	6D	7E	0C
6	67	D9	80	59	05	49	06	D4	6F	9D	8A	78	34	AB	17	B6
7	9C	41	A3	57	FB	69	C9	FF	3E	EE	B8	D1	18	75	1B	A0
8	51	32	3A	E5	43	6B	E9	8C	09	07	D0	36	1D	BE	1E	A8
9	35	E4	70	89	55	C8	B5	27	BF	CF	84	A5	16	1C	DF	3B
A	12	F9	82	2F	AC	94	5A	76	47	4B	79	C5	7C	7D	68	0E
B	A4	ED	B7	A1	F8	4A	03	8F	46	91	39	14	08	E3	4D	81
C	1F	23	37	9A	7B	B9	50	60	92	F5	73	6C	D5	FE	42	F6
D	53	E8	87	3F	38	90	A7	52	D3	21	E6	B1	01	CD	C0	96
E	D8	2B	F7	86	F4	A9	5D	4C	25	F0	40	71	4E	F2	FC	45
F	22	5E	72	C7	DC	5B	5F	3C	E7	65	61	6E	0D	C6	44	9E

Figure 2. Possible S-BOX created with the help of K/DSA [13]

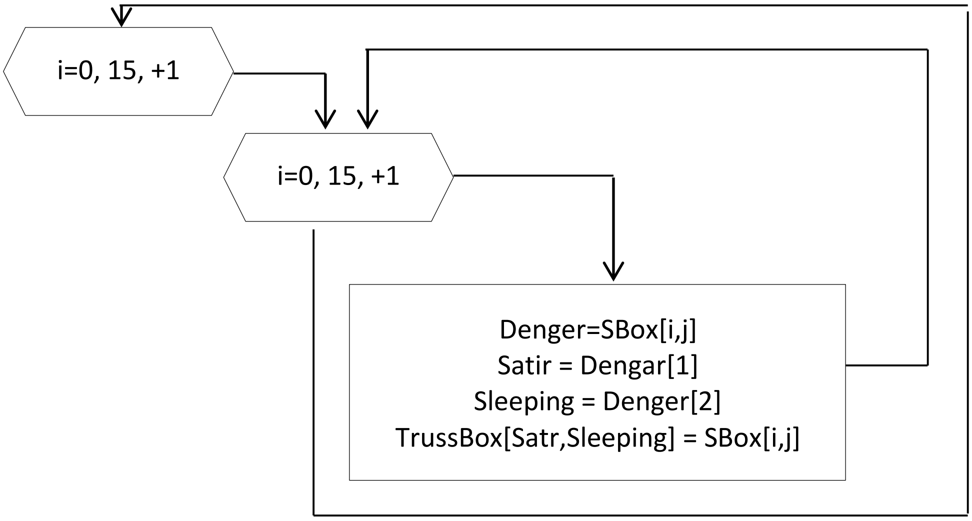
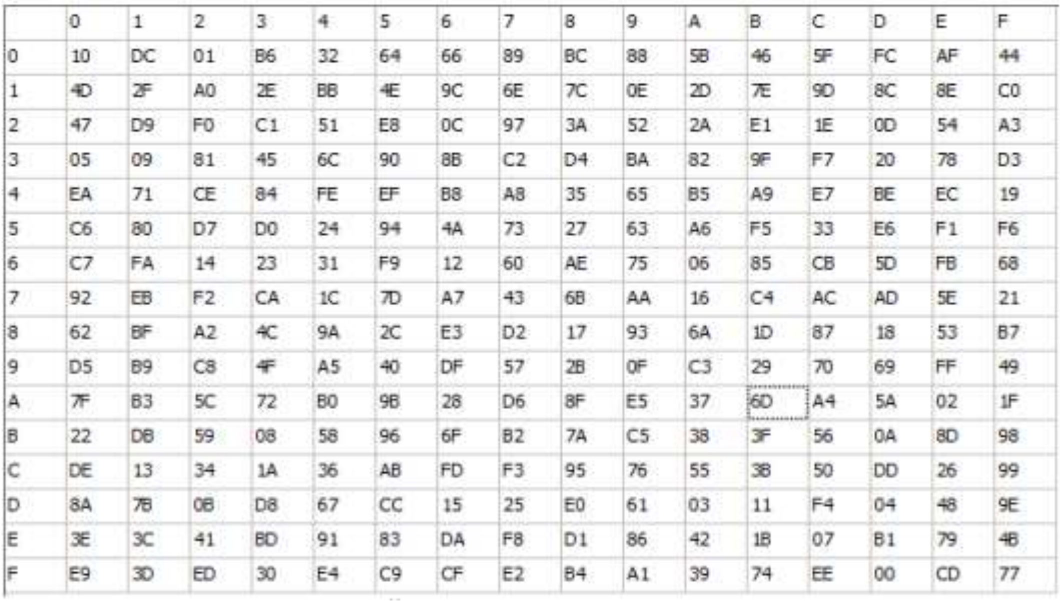


Figure 3. Obtaining an inverted S-Box [13]

The resulting S-box is used to produce the inverted S-box. The inverted S box is obtained as follows: For example, in the S-box shown in Figure 2, the value of AB is at the intersection of the sixth row and column D. The value AB will be accepted as column A. Row B. In the inverted S-box, the value in S-box, 6D, will be written at this intersection. This process is applied to all values in the S-box. Getting the dynamically inverted S-box is an important factor in reducing the decoding time. A possible inverted S-box is shown in Figure 4.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	10	DC	01	B6	32	64	66	89	BC	88	5B	46	5F	FC	AF	44
1	4D	2F	A0	2E	BB	4E	9C	6E	7C	0E	2D	7E	9D	8C	8E	C0
2	47	D9	F0	C1	51	E8	0C	97	3A	52	2A	E1	1E	0D	54	A3
3	05	09	81	45	6C	90	8B	C2	D4	BA	82	9F	F7	20	78	D3
4	EA	71	CE	84	FE	EF	B8	A8	35	65	B5	A9	E7	BE	EC	19
5	C6	80	D7	D0	24	94	4A	73	27	63	A6	F5	33	E6	F1	F6
6	C7	FA	14	23	31	F9	12	60	AE	75	06	85	CB	5D	FB	68
7	92	EB	F2	CA	1C	7D	A7	43	6B	AA	16	C4	AC	AD	5E	21
8	62	BF	A2	4C	9A	2C	E3	D2	17	93	6A	1D	87	18	53	B7
9	D5	B9	C8	4F	A5	40	DF	57	2B	0F	C3	29	70	69	FF	49
A	7F	B3	5C	72	B0	9B	28	D6	8F	E5	37	6D	A4	5A	02	1F
B	22	DB	59	08	58	96	6F	B2	7A	C5	38	3F	56	0A	8D	98
C	DE	13	34	1A	36	AB	FD	F3	95	76	55	3B	50	DD	26	99
D	8A	7B	0B	D8	67	CC	15	25	E0	61	03	11	F4	04	48	9E
E	3E	3C	41	BD	91	83	DA	F8	D1	86	42	1B	07	B1	79	4B
F	E9	3D	ED	30	E4	C9	CF	E2	B4	A1	39	74	EE	00	CD	77

Figure 4. Production of a possible inverted S-box [22]

Only one Flipped S-Box can be obtained from each S-Box. Getting the Flipped S-Box depends on creating the S-Box.

2.3 Images Cryptographic Process

Given today's technology, it is inevitable that image encryption will be applied to color images. Color images are composed of three primary colors, RGB (Red, Green, and Blue). With the RGB space, all colors in nature can be obtained by mixing these three primary colors in certain proportions. In this method, color images in bmp and jpg formats with a depth of 24 bits are used. In the first step of the encryption process, the RGB values of each pixel are obtained from the original image. The hexadecimal equivalent is taken for each color value. The first value of the hexadecimal equivalent obtained is accepted as a row, and the second value is accepted as a column. The value at the intersection of the row and column values in the S-box obtained with K/DSA is processed as new color values. This process is applied separately to the RGB values of each pixel. For example, suppose the RGB values for a pixel are R=59, G=167, and B=218. The hexadecimal equivalents are R=3B, G=A7, and B=DA. Given a possible S-box produced with K/DSA in Fig. 2, new R = CB for row 3, column B, new G = 76 for row A, column 7; new B = E6 for row D, column A. Acquired. The method examined is more robust; the newly obtained RGB values are XORed with a hexadecimal value at random coordinates in the S-box. A pseudo-random number generator (PRNG) is used to generate the random numbers.

PRNGs are types of algorithms that produce a series of numbers that are not easily associated with their elements [13]. The main purpose of these number generation methods is actually to generate a series of numbers based on a particular mathematical function. By hiding the mathematical function used during number generation, an attempt is made to prevent estimation of the numbers produced. In number generation using PRNG, the same number sequence is always obtained with the help of an initial value. Therefore, the base value specified by the user is used as the initial value of the PRNG used in the method. Two hexadecimal numbers are generated by PRNG to represent the row and column for each RGB value. Since the initial value is the user's private key, the coordinates of the S-box generated by PRNG will always be the same. Specifying a different user key will result in different coordinates. The new RGB values after the XOR operation represent the new pixel values of the encoded image. The flowchart performing the encryption process to represent the original image before encryption is shown in Figure 5.

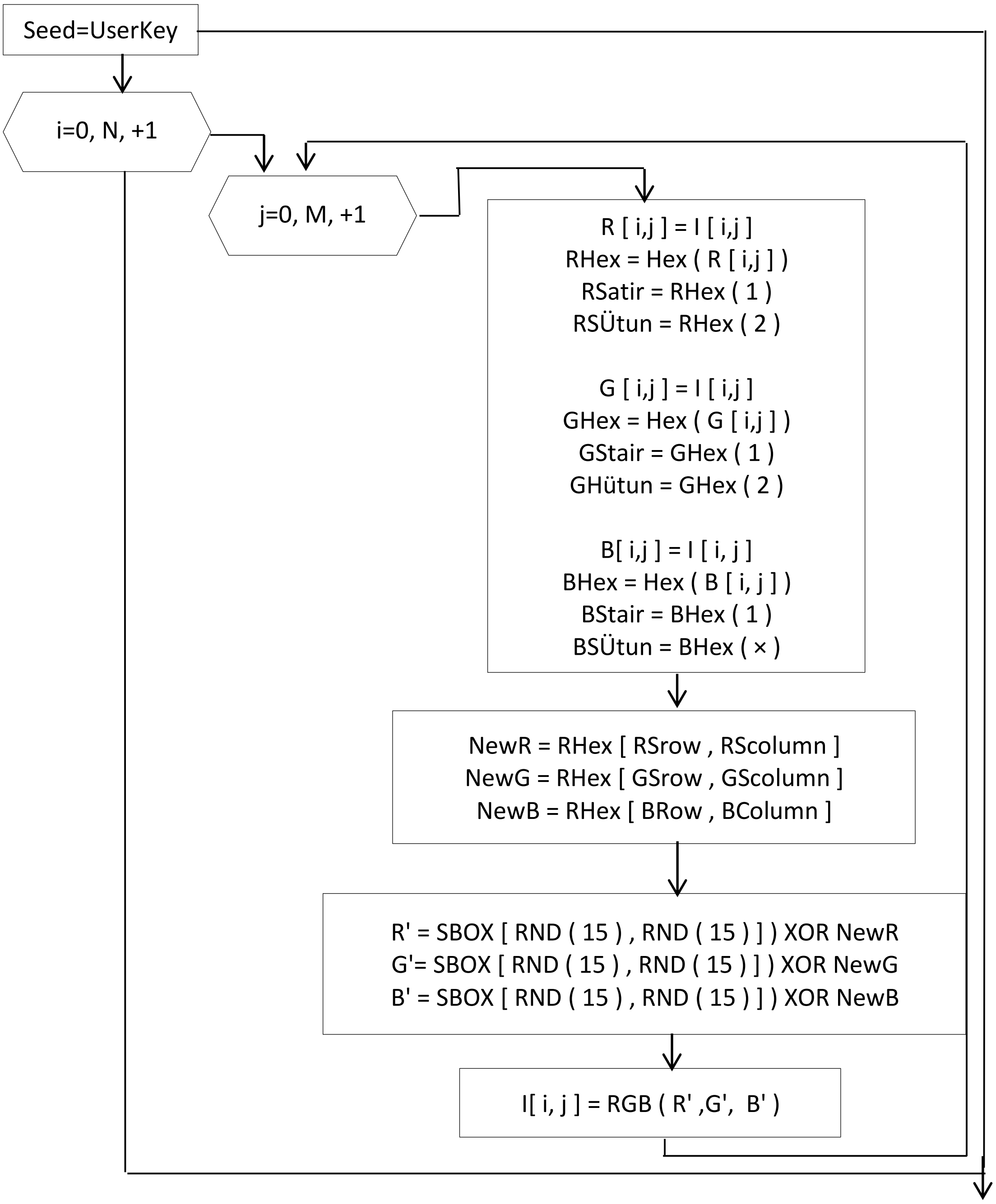

Seed=UserKey
i=0, N, +1
j=0, M, +1
R [i,j] = I [i,j]
RHex = Hex (R [i,j])
RSatir = RHex (1)
RSÜtun = RHex (2)
G [i,j] = I [i,j]
GHex = Hex (G [i,j])
GStair = GHex (1)
GHütun = GHex (2)
B[i,j] = I [i, j]
BHex = Hex (B [i, j])
BStair = BHex (1)
BSÜtun = BHex (×)
NewR = RHex [RSrow , RScolumn]
NewG = RHex [GSrow , GScolumn]
NewB = RHex [BRow , BColumn]
R' = SBOX [RND (15) , RND (15)]) XOR NewR
G'= SBOX [RND (15) , RND (15)]) XOR NewG
B' = SBOX [RND (15) , RND (15)]) XOR NewB
I[i, j] = RGB (R' ,G', B')

Figure 5. Flowchart of the encryption process

The same random numbers will always be generated unless the user key is changed. This operation is performed separately for each value of R, G, B. The hexadecimal value in the S box and the R, G, and B values obtained from the encrypted image are XORed. For each of the new R, G and B values obtained after the XOR operation, the points where the corresponding row and column values intersect in the inverted S-box provide the pixel value of the original image. For example, suppose the new value of R obtained after the XOR operation is AB. Looking at row A and column B in the inverted S-box in

Figure 4, it will be seen that the corresponding value is 6D. When these operations are applied to the R, G, and B values of each pixel after the XOR operation, the pixel values of the original image are reached. The flowchart illustrating the decoding process to represent the encoded image I is shown in Figure 6.

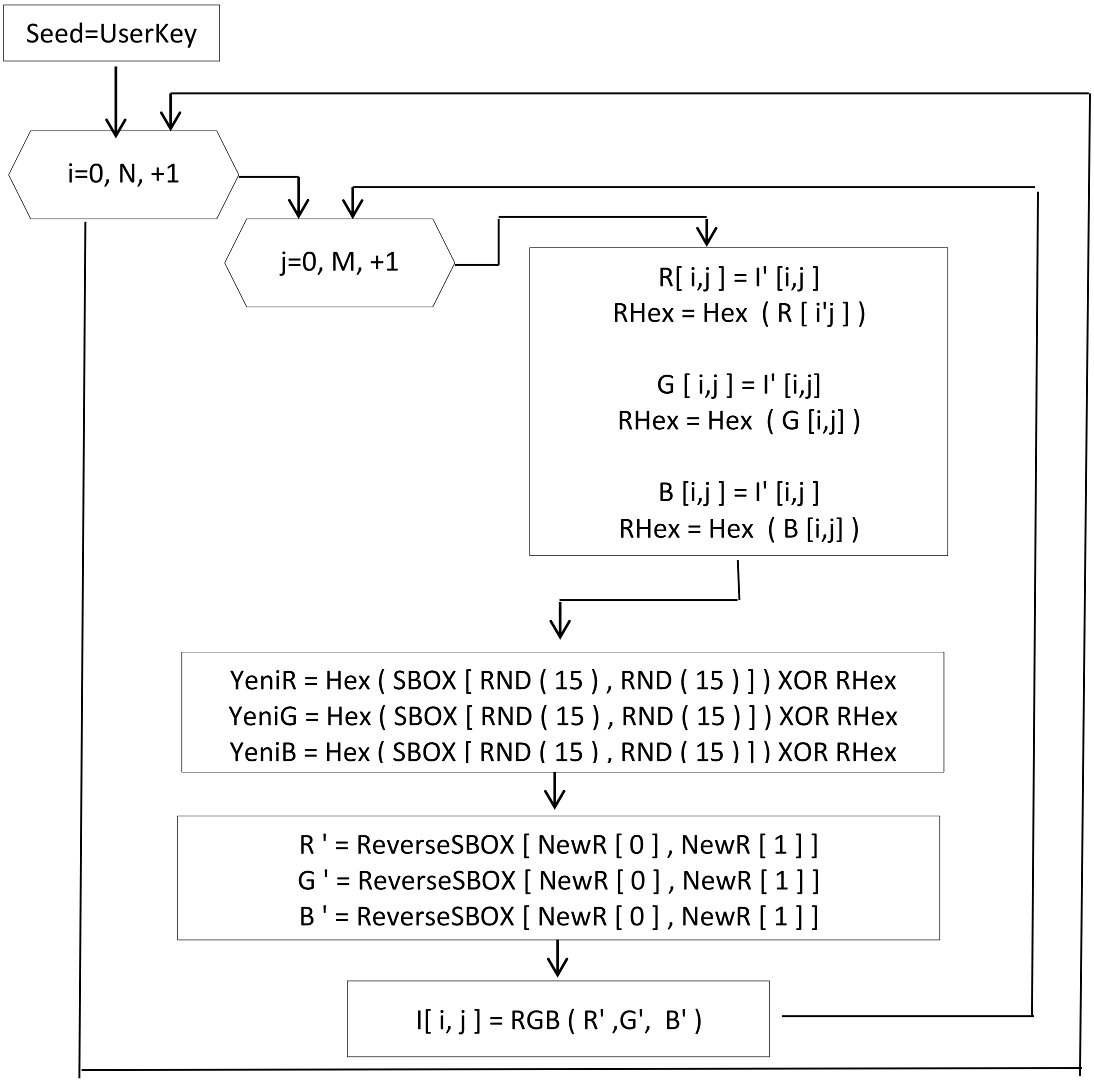

Seed=UserKey
i=0, N, +1
j=0, M, +1
R[i,j] = I' [i,j]
RHex = Hex (R [i'j])
G [i,j] = I' [i,j]
RHex = Hex (G [i,j])
B [i,j] = I' [i,j]
RHex = Hex (B [i,j])
YeniR = Hex (SBOX [RND (15) , RND (15)]) XOR RHex
YeniG = Hex (SBOX [RND (15) , RND (15)]) XOR RHex
YeniB = Hex (SBOX [RND (15) , RND (15)]) XOR RHex
R ' = ReverseSBOX [NewR [0] , NewR [1]]
G ' = ReverseSBOX [NewR [0] , NewR [1]]
B ' = ReverseSBOX [NewR [0] , NewR [1]]
I[i, j] = RGB (R' ,G', B')

Figure 6. The flowchart illustrating the decoding process to represent the encoded image

3. Results and Discussion

This section discusses the security analysis of the method detailed above. For this purpose, the results obtained by performing graph, correlation, and principal space analyses are presented below.

3.1 Correlation Analysis

Statistical correlation is a measure of the strength of a linear relationship between two random variables. In an array containing n elements, where x and y are random variables, the correlation coefficient can be calculated with the following equation [14].

$$=\frac{cov(x.y)}{\sqrt{D(x)D(y)}}r_{xy} \quad (2)$$

from here;

$$cov(x.y)=\frac{1}{n}\sum_{i=1}^{n}[x_i-E(x)][y_i-E(x)] \quad (3)$$

$$D(X)=\frac{1}{n}\sum_{i=1}^{n}[x_i-E(x)]^2 \quad (4)$$

$$E(X)=\frac{1}{n}\sum_{i=1}^{n}x_i \quad (5)$$

In this study, horizontal, vertical, and diagonally adjacent pixels are taken into account to calculate the correlation of pixels in the encoded image. For both the original and the encoded images, 2000 random pixels were selected in each direction (horizontal, vertical, and diagonal) and adjacent to each other. In order to evaluate the performance of correlation analysis, these processes were applied to the "baboon.bmp" and "landscape.jpg" images. The correlation calculation results obtained for each R, G, and B color layer are shown in Table 1 and Table 2.

Table 1. Correlation coefficients for "baboon.bmp"

		Blue	**Green**	**Red**
The original picture	Horizontal	0.91556	0.85555	0.89887
	Vertically	0.87001	0.80001	0.900111
	Diagonally	0.79993	0.74221	0.90001
Encrypted image	Horizontal	-4.9268x10^{-5}	-0.0035	2.2763x10^{-4}
	Vertically	-0.0027	-0.0010	0.0012
	Diagonally	-0.0014	-0.0011	-0.0034

Table 2. Correlation coefficients for "Landscape.jpg"

		Blue	**Green**	**Red**
The original image	Horizontal	0.9866	0.9791	0.9696
	Vertically	0.9809	0.9720	0.9595
	Diagonally	0.9764	0.9651	0.9491
Encrypted image	Horizontal	7.4524x10^{-4}	-0.0015	-5.1060x10^{-5}
	Vertically	6.4401x10^{-4}	-0.0012	-6.3353x10^{-4}
	Diagonally	-8.1701x10^{-4}	-0.0015	-4.5858x10^{-4}

The correlation coefficient is supposed to carry values confined between one positive and one negative (an integer). If this is the case, this means that the correlation between pixels is strong. If the correlation is close to absolute zero, this indicates a weak pixel correlation.

From the results and figures of the following two tables, it is clear that:

- The correlation between the main images is close to one.
- The correlation between the coded images is close (to a very high degree) to zero.

Conclusion: ((The method of encrypting the image using S-box makes the relationship between neighboring pixels close to 0 and gives successful results).

3.2 SSIM Structural Similarity Test

This test is a guide or method for measuring and determining the degree of similarity between images. You can also set the quality (image quality) as well. The following equation illustrates this [15, 16].

$$S(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \qquad (6)$$

To interpret the results, it can be said:
- The two pictures are similar = the value is one or closes to it.
- There is no similarity = zero value or close to it.

The following table demonstrates this:

Table 3. Test results /Structural similarity

Picture	percentage	milarity percentage between nal image and encrypted image
Baboo	0.01478	1.0001
Manzara	0.01217	1.0001

From Table 3, we can conclude the following:
- The similarity between the image (which has already been decrypted) and the original images is 100%.
- The percentage of similarity between the original images and the encrypted image is very low.
- No loss of pixels in the decrypted process.

3.3 Entropy Test

Entropy is the randomness and disorder in a system. It is calculated according to the equation below:

$$\mathrm{H(m)} = \sum_{i=0}^{MxN-1} p(m_i) log_2 \frac{1}{p(m_i)} \qquad (7)$$

Here, $p(m_i)$ represents the m_i probability states of tokens in a message, and M × N represents the total number of tokens [17, 18]. In a random message, the ideal entropy value is 8, and in messages with less randomness, the entropy value is less than 8. If the entropy value is much less than 8, a security threat can be expected [16].

Given that the encryption images are composed of random pixel values, the ideal entropy value is expected to be 8. The entropy test can be applied to gray-level images. In color images, the entropy test is applied separately for each RGB channel. Table 5 shows the entropy test results for the original, encoded, and decoded RGB color images.

Table 4. Entropy test results

	Entropy of an original image			Entropy of encrypted image			Entropy of decrypted image		
	R	G	B	R	G	B	R	G	B
Baboo	7.6998	7.5002	7.6987	7.9992	8	8	7.6999	7.5345	7.6899
Manzara	7.8001	7.8999	7.4001	7.9996	7.9876	8	7.7899	7.9732	7.4123

*R-Red; G -green; B- Blue

Looking at Table 5, it is evident that the entropy values of the encrypted images are very close to 8. Therefore, it can be said that the method is quite resistant to attacks. In addition, the fact that the entropy values of the original and decrypted images are the same indicates that there is no data loss in the encryption and decryption processes.

4. Conclusion

In this study, dynamic encryption techniques are explored and studied to encrypt and decrypt images using dynamically generated S-boxes. The S-boxes are dynamically generated based on the user key. When different user keys are used, different S-boxes are dynamically obtained. Therefore, it has a very large key area. Since the encryption and decryption processes are based on S-boxes, obtaining the original image will only be possible if you know the correct user key. In addition, when the PRNG equation to be used in the method is kept secret, the probability of obtaining the original image is greatly reduced. It is very important that the original image can be recovered from an encrypted image. Considering the experimental results, it has been observed that the method allows restoring the original image with the correct key value without any data loss. Consistent with the results obtained from all the experimental results, it seems possible to use the method for the purpose of image encryption.

References

1. Ge, Xinrui; Jia Yu; Hanlin Zhang; Chengyu Hu; Zengpeng Li; Zhan Qin; Rong Hao*T ,owards achieving keyword search over dynamic encrypted cloud data with symmetric-key based verification. IEEE Transactions on Dependable and Secure computing*, **2019**. *18(1):* p. 490-504.
2. Fernando, Erick; Dine Agustin; Muhamad Irsan; Dina Fitria Murad; Hetty Rohayani; Dadang Sujana. *Performance comparison of symmetries encryption algorithm AES and DES with raspberry pi*. in *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*. **2019**. IEEE.
3. Shukur, W.A., *A proposed method for generating a private key using digital color image. International Journal of Applied Engineering Research*, **2017**. *12(16):* p. 6235-6240.
4. Sankpal, P.R.; Vijaya, P.A., "Image Encryption Using Chaotic Maps: A Survey", Fifth International Conference on Signal and Image Processing, 8-10 January **2014**, *Jeju Island*, pp.102-107, DOI: 10.1109/ ICSIP.2014.80.
5. Kubba, Z.M.J.; Hoomod, H.K., *A Hybrid Modified Lightweight Algorithm Combined of Two Cryptography Algorithms PRESENT and Salsa20 Using Chaotic System*. in *2019 First International Conference of Computer and Applied Sciences (CAS)*. **2019**. IEEE.
6. Kubba, Z.M.J.; H.K. Hoomod. *Developing a lightweight cryptographic algorithm based on DNA computing*. in *AIP Conference Proceedings*. **2020**. AIP Publishing LLC.

7. Abraham, L.; Daniel, N., “Secure Image Encryption Algorithms: A Review”, *International Journal of Scientific & Technology Research,* (**2013**) *2(4)* 186-189.

8. Sheng, T.; Hong, Q.; Junjie, X., *K-DSA* for the multiple traveling salesman problem. *Journal of Systems Engineering and Electronics,* **2023**.

9. Moses, L.E.; Oakford R.V., "Tables of Random Permutations", Stanford University Press, ISBN-13: 978-0804701488, **1963**.

10. Durstenfeld, R., "Algorithm 235: Random permutation", *Communications of the ACM*, (**1964**) *7(7)* 420.

11. Knuth, D.E., "The Art of Computer Programming", 2th Edition, Addison-Wesley, **1969**, 139- 140.

12. Güvenoğlu, E.; Esin, E.M., "Knutt/Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması", *Politeknik Dergisi,* (**2009**) *12(3)* 151-155.

13. Shukur, W.A.; Qurban, L.K. ; Aljuboori, A., Digital Data Encryption Using a Proposed W-Method Based on AES and DES Algorithms. *Baghdad Science Journal*, **2023**.

14. Özcan, Hikmetcan; Fidan Kaya Gülağiz; Mehmet Ali Altuncu; SüMeyya İlkin; Suhap Şahin, A new visual cryptography method based on the profile hidden Markov model. *Advances in Electrical and Computer Engineering*, **2021**. *21(1):* p. 21-36.

15. Hussain, I.; Shah, T.; Gondal, M.A.; Mahmood, H., " Efficient method for designing chaotic S-boxes based on generalized Baker's map and TDERC chaotic sequence", Nonlinear Dynamics, (**2013**) *74(1)* 271-275.

16. Osorio, Felipe; Ronny Vallejos; Wilson Barraza; Silvia María Ojeda; Marcos Alejandro Landi, Statistical estimation of the structural similarity index for image quality assessment. Signal*, Image and Video Processing*, **2022**: p. 1-8.

17. Naveenkumar, S.K.; Panduranga, H.T., "Triple image encryption based on integer transform and chaotic map", *International Conference on Optical Imaging Sensor and Security (ICOSS),* 2-3 July **2013**, *Tamil Nadu, India*, pp. 1-6.

18. Simion, E., Entropy and randomness: From analogic to quantum world. *IEEE Access,* **2020**. 8: p. 74553-74561.