Ali, AH, Gbashi, EK, Alaskar, H and Hussain, AJ

A Lightweight Image Encryption Algorithm Based on Secure Key Generation

Article

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 April 2024, accepted 9 June 2024, date of publication 11 July 2024, date of current version 19 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3414334

RESEARCH ARTICLE

A Lightweight Image Encryption Algorithm Based on Secure Key Generation

A'LAA HUSSEIN ALI[1], EKHLAS KHALAF GBASHI[1], HAYA ALASKAR[2], AND ABIR JAAFAR HUSSAIN[3,4], (Senior Member, IEEE)

[1]Department of Computer Science, University of Technology, Baghdad 10011, Iraq
[2]Computer Science Department, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[3]Department of Electrical Engineering, University of Sharjah, Sharjah, United Arab Emirates
[4]School of Computing and Mathematical Sciences, Liverpool John Moores University, L3 3AF Liverpool, U.K.

Corresponding author: Abir Jaafar Hussain (abir.hussain@sharjah.ac.ae)

This work was supported in part by the Department of Electrical Engineering, University of Sharjah, under Grant 22020403199; and in part by the Prince Sattam Bin Abdulaziz University under Project PSAU/2024/1445.

ABSTRACT Data confidentiality and security are important issues due to the sensitivity of the data and its relationship with users' privacy. Sensitive data includes images and texts that can be transmitted over the Internet, Internet of Things devices and edge-fog-cloud system. These devices require speed and accuracy responses, and they are vulnerable to hacking. To solve these problems, encryption algorithms provide necessary solution to meet these requirements. Advanced Encryption Standard represents the best development in data encryption; however, it is computational expensive. In this research, an improved advanced encryption standard algorithm is proposed with advanced security and lightweight computation utilized for encrypting of images and texts. The algorithm is improved using various steps including key generation which is performed in two steps. First, using an innovative, proven chaotic function distinguished by its sensitivity to any change in its variables. Second, using three-dimensional Lorenzo function. In our research, a unique key was used for all rounds, and round key then used like advanced encryption standard. Two new dynamic substitution boxes are used one for odd rounds and the other for even rounds in which the speed does not exceeding a millisecond. The mix column function was replaced by a circular permutation function at the bit level, which improved the speed and performance of the algorithm, Our extensive simulation results indicated enhanced speed, randomness, and high efficiency in encrypting Internet of Things data. The algorithm was evaluated using the National Institute of Standards and Technology tests.

INDEX TERMS AES, chaotic map, circular permutation, cryptography, dynamic S-box, IoT, lightweight algorithm.

I. INTRODUCTION

Internet of Things (IoT) has transformed industries, smart homes, and healthcare, but it has also brought up new security challenges, especially for low-resource devices [1]. Fog computing increases data security and privacy; however, its high processing requirements and its drawbacks with traditional methods necessitate lightweight cryptography [2]. Advanced encryption standard (AES) algorithm is widely utilized due to its exceptional dependability and applicability in various applications [3]. IoT devices struggle to handle AES algorithm due to its complexity [4], [5]. Low-powered IoT devices can benefit from lightweight algorithms since they balance security and resource efficiency [6]. The main purpose of data encrypting is to prevent unauthorized access during transmission and storage [7]. For defensive purposes, encryption technique or system must be nearly unbreakable [8]. Pseudo-random key sequences have been generated for processing data encryption and decryption [9]. The chaotic sequence can theoretically be made unbreakable by utilizing pseudo-randomness and predictive complexity of chaotic systems and their sensitivity to initial values to

The associate editor coordinating the review of this manuscript and approving it for publication was Leandros Maglaras.

produce the same encryption effect as a random key in a one-time pad [10], [11]. Because of the differences in the computational costs, encrypting image formats presents security risks, especially when delivering images via wireless networks [12]. This research aims to create an effective, lightweight, fast, and safe encryption framework for images (gray and color) and text byg safeguarding important data within these resource-limited devices' networks. Our goal is to create a more reliable and secure IoT utilizing the combined capabilities of AES improvement and chaos-based key generation in the Fog computing architecture.

In particular, the novelty of the proposed works includes the following:

- Chaos-based key generation: as far as the authors are concerned, this is the first-time chaotic maps are utilized to generate pseudo-random key streams for image and text encryption, taking advantage of their inherent resistance to statistical analysis and sensitivity to initial conditions.
- AES enhancement for IoT: improving AES algorithm's performance for image encryption on IoT devices with limited resources while preserving a high level of security.

Simulation results indicated that the proposed approach is faster at encrypting data when benchmarked with state-of-the-art algorithms. Furthermore, it is demonstrated that the throughput of the proposed method is 32,803.69 for encryption and 92,619.39 for decryption. While the algorithm nonlinear characteristic is maintained, the recommended modifications did not weaken the security of the conventional AES. Cipher attacks require extensive amount of time to break the proposed algorithm.

The reminder of this paper is organized as follows. Section II provided extensive information about the literature review. The suggested work and associated approach are covered in Section III. The results and analysis of the proposed method are shown in Section IV, along with benchmark with state-of-the-art techniques. Sections V and 6 show the results and the conclusion, respectively.

II. RELATED WORK

This section discusses the development of secure keys and lightweight image encryption technique.

The authors in [13] combined Henon map with AES algorithm in which AES algorithm encrypts plain image, while the Henon map generates a random key for the encryption stage, enhancing security and withstanding more attacks. The results showed that their approach performs satisfactorily when encrypting images [13].

The authors in [14] introduced a technique based on AES to enhance the S-box generation, in which they claim that their algorithm demonstrates its effectiveness in generating encrypted messages with a larger average avalanche impact when benchmarked with the original AES method. Avalanche effect shows that the outcomes of encrypted texts by the modified S-box is 51% on average showing an improvement of 3% on average when benchmarked with the original AES method.

In [15], the authors introduce a novel AES encryption S-box with superior distance to Strict Avalanche Criterion (SAC), Bit Independence Criterion (BIC), and algebraic complexity, enhancing security measures by evaluating key cryptographic parameters. The authors claimed that their proposed S-box performs at least as well as the standard S-boxes.

In [16], an enhanced AES algorithm incorporates dynamic S-box creation and key generation was proposed. The authors indicated that their algorithm enhances security against attacks in online transactions, banking, and e-commerce by adding complexity to the cipher text.

A keystream generation method utilizing the Chebyshev map was proposed in [17], the algorithm offers real-time image encryption with strong cryptographic strength, meeting NIST requirements. Only a small portion of a minute is needed for the application, which is considered a long time.

The enhanced AES technique proposed in [18], uses a 256-bit random number generator to generate a randomized S-box, enhances security in military applications and IoT security with low overheads.

In [19], Small Lightweight Cryptographic Algorithm (SLA), a lightweight encryption technique based on the substitution permutation network (SPN), offers faster security than the Feistel-based ciphers, making it suitable for embedded settings like Radio-Frequency Identification (RFID) tags and wireless sensor nodes.

Based on chaotic theory, a first phase of permutation has been added to the AES algorithm to improve confusion rates was proposed in [20]. Two DNA sequences are used to create the mixing matrix from the logistic map 3D; the first sequence serves as a key in the add round key. The decryption output is a DNA sequence sent to the recipient. Completing the encryption and decryption procedures after five statistical and NIST checks took a few seconds.

The research proposed in [21] creates S-box using shifting, a circle map, and a password key, highlighting its dynamic nature, input-output interaction, and intricate production process.

Table 1 shows various techniques for data encryptions with their advantages and disadvantages. As it can be noted, majority of these techniques suffer from low speed, complexity, and low security. The aim of this work is to overcome these issues by using lightweight algorithm with improved process complexity while preserving the security of the data.

III. BACKGROUND AND METHODOLOGY

In this section, basic background about the well-known Advanced Encryption standard as well as our proposed methodology will be provided.

TABLE 1. Benchmarking various techniques for data encryptions.

Reference	Improvement	Advantages	Disadvantages	Quality measure (time, security)
[13]	• Encrypt images using the AES algorithm. • Generate a random key by utilizing the Henon map. • XOR operation between the results of the first and a second steps is used.	• Key is sensitive to any modification • Employing AES encryption and the • XOR operation gives the method its enhanced level of safety.	• AES is a reasonably secure algorithm, but it cannot be used to encrypt photographs. • It is complex and it is time consuming because the data associated with the images are much larger and include a higher degree of redundancy than text data.	• The method is highly safe and low speed.
[14]	• Enhanced AES S-box generation by combining its original S-box • S-box is generated by Logistic Map.	• Proposes method to produce the S-box. • Modified S-box scored higher average of avalanche effect compared to the original method.	• The method does not change the process on how AES works but rather proposes another method to produce the S-box.	• The results of the method do not differ from the original algorithm, • It has high complexity and time consumption.
[15]	• Introduce a way to create S- box for AES	• The performance of the new S-box is at least as good as the performance of the existing S-boxes. • The proposed S-box has better distance to SAC, BIC and algebraic complexity.	• The time it takes to generate S-box is longer compared to the original algorithm.	• It has more security but more consumption time • It is not suitable for low resource devices.
[16]	• Enhance AES algorithm with hybrid approach of Dynamic Key and S-box Generation.	• AES algorithm with hybrid approach provides strong security in message transmission by adding more complexity in AES to increase Confusion and Diffusion in Cipher text. • It protects message from Brute-force Attack, Differential	• Requires high computational resources.	• Effective technique for the applications based on Internet.

TABLE 1. *(Continued.)* **Benchmarking various techniques for data encryptions.**

		Attack, Algebraic Attack and Linear Attack. • Effective technique for the applications which are based on internet.		
[17]	• Use a type of chaotic map like the Chebyshev 1D map, based on the parameters, for a good random appearance. • The output is a test of several measurements, including complexity balance and time execution.	• Results show that any change in input will change the output. • It has good coding strength with added benefit that resists security analysis.	• Time required for the application is parts of the minute • Considered long time for real-time image encryption.	• The algorithm has good coding strength with added benefit that resists security analysis and requires only parts of the minute considered long time.
[18]	• Enhance AES algorithm with a randomized S box.	• Provides a higher degree of security compared to original AES due to the utilized randomized S-box.	• Claiming that the overhead is an additional 0.005 seconds taken to generate the random number and re-arrange the S-Box that make the proposed method lower than the original AES.	•It has high degree of security and long time to execution •Complex computation like the standard AES.
[19]	• Ultra-lightweight encryption scheme. • It relies on substitution permutation network (SPN). • It uses 64-bit plaintext and supports a key length of 80/128-bits. • It has nonlinear layers, XOR operations, and round permutation layers.	• Lightweight cryptographic scheme, • Construction of a strong non-linear S-box (confusion layer) uses Galois field multiplication which meets cryptographic properties and provides a novel method to construct diffusion layers by 32-bit binary matrix. • The scheme provides a sufficient security level against most of the well-known attacks on block ciphers, such as	• Half of the ciphertext bits are affected by a single bit change in the proposed cipher's key. • This is not good, as the effect must spread to all ciphertext.	•Has highest throughput of 97,050.44167 kilobytes per second •High speed and enough security.

TABLE 1. ***(Continued.)*** **Benchmarking various techniques for data encryptions.**

		linear and differential cryptanalysis.		
[20]	• Enhancement AES based on 3D Chaos Theory and DNA Operations • A new step added to the AES's four stages of permutation. • Chaos theory is adopted in the permutation stage and the shift phase. • DNA adopted at the phase of the mix and add round key stage due to its high storage capacity.	• Increasing the random and break the link between the plaintext and encrypted texts by increasing the confusion and diffusion through the stages of the system.	• A new step added to the AES's four stages of permutation. • Increased the computational complexity and time consumption because the standard AES is high complex and time consumption.	• The time in the encryption and decryption in the enhancement algorithm takes a few milliseconds.
[21]	• AES S-Box generation Based on user's Password Key and 1D Circle Map.	• Using dynamic S-Box that eliminates AES security risks.	• AES S-box Generation is an important stage, and in this method it depends on the key. • If the key is revealed, the security of the entire algorithm will be threatened.	• A few milliseconds are needed to implement S-box generation and enhance the original AES S-box algorithm security.
Proposed work	• Enhance the AES algorithm in three stages key generation based novel chaotic map, S-box generation based key, replacing mix column function.	• Improving the algorithm in terms of safety by making the S-box dynamic • Improving its speed by replacing the mix column function that consumes time and computation and making it suitable for IoT devices.	• The proposed work has been implemented to encode text, color, and grayscale images. • It will be tested for video encryption in the future.	• The time taken for encryption and decryption is considered very short compared to the state-of-the-art methods. • The algorithm has also been improved to increase its security and make it suitable for IoT devices.

A. ADVANCED ENCRYPTION STANDARD ALGORITHM (AES)

The AES algorithm, a symmetric key algorithm, established by the US National Institute for Standard and Technology (NIST) as the standard for digital data encryption method [22]. AES comprises three types, which are: AES-128, AES-192, and AES-256, each with varying key sizes that determine the allocation of rounds [23], [24]. AES's functions are Add-Round-Key, Shift-Rows, Mix-Columns, and Sub-Bytes [25], [26]. The 128-bit AES algorithm uses 08 rounds of transformations, including Add Round Key, Sub Bytes, Shift Rows, and Mix Columns, with round 10 having all transformations except for Mix Column, and the decryption process is the exact inverse.

B. PROPOSED METHOD OF LIGHTWEIGHT AES

This work proposes a lightweight, fast, and secure improved AES encryption method for IoT sensor data security called simple swift IoT guard (ss IoT g). It creates secret keys using an innovative map and Lorenzo 3Dimension. The algorithm

Algorithm 1 Key Generation by the Proposed Map

Inputs: p, x, n: The parameter 'p,' the most important variable in deciding how the logistic map behaves; x: The initial value, n: The number of iterations to generate the key
Output: key: A byte array containing the generated key
- Loop 'n' times:
Step1: Run the new chaotic equation for n steps and define

$$f(x) = \left(\frac{(x \times r) + (1 - x)}{0.9 \times r}\right) \%1$$

Step 2: If 'x' is larger than 0.5, put '1' to 'binary string'; Else, append '0'.
Step 3: If the length of the 'binary string' reaches 8:
a. Convert 'binary string' to an integer value.
b. Add the integer value (byte) to the 'key' byte array.
c. Reset 'binary string' to an empty string for the next set of bits.
Step 4: Return the generated 'key' byte array.

generates two S-boxes and uses circular permutations instead of mixed columns. Figure 1 shows our proposed algorithm.

As illustrated in Figure 1, our proposed algorithm has the following improvements.

1) KEY GENERATION OF SYSTEM

key generation is performed using two proposed methods.

a: A NOVEL CHAOTIC MAP FOR KEY GENERATION

This work presents a new chaotic map that has been derived from logistic maps and experiments, which allows the development of a chaotic pseudo-random number generator.

The generated pseudo-random numbers pass randomness tests with uniform distribution. A technique for s-box production is proposed, using an encryption key with a high correlation with s-box generation.

Expanding the map enhances system complexity and key space, improving cryptosystem security. Additional statistical analyses and computer simulations validate the suggested map's high level of security.

The proposed function that generates random pseudo number is define as follows.

$$f(x) = \left(\frac{(x \times r) + (1 - x)}{0.9 \times r}\right) \%1 \tag{1}$$

where x is the generated values selected between 0 and 1, r is the initial value.

Algorithm 1 shows the full key generation process using the proposed f function.

b: LORENZO MAPS 3D FOR KEY GENERATION

A chaotic map in general has three categories of dimensions. The 3D-Lorenzo chaotic map approach uses three variables to represent the geometric position of points in space [28]. In our study, this approach is employed due to its simplicity, allowing the utilization of the user input, and generating accurate output data as follows.

$$x = p \cdot (x - y), \tag{2}$$
$$y = (x.z) + (r.x) - y, \tag{3}$$
$$z = (x.y) + (t.z). \tag{4}$$

where x, y and z represent the input values p, r, t are parameters with p = 10, r = 28, t = 8/3 while x_1, x_2 and x_3 are the input values.

Algorithm 2 shows the process of generating security keys using the proposed 3D-Lorenz map.

Algorithm 2 Generating Key Using the Lorenzo Map

Inputs: p, r, t (parameters for the Lorenzo model), x_{old}, y_{old}, z_{old} (initial random values within the range [0, 1].), n: The number of iterations to generate the key
Output: key: A byte array containing the generated key
Loop 'n' times
Begin
Run Lorenzo Model for n Steps:
x_{new} = p. (x − y),
y_{new} = (x. z) + (r. x) − y,
z_{new} = (x. y) + (t. z)
Return the Key
End

2) AES ENHANCEMENT

This study presents a novel method for creating improved S-boxes with enhanced cryptographic features, including bit independence criteria, periodicity, algebraic complexity, strict avalanche criteria, and distance to SAC. Furthermore, it substitutes circular permutation at the bit level for the mix column function of the conventional AES algorithm.

a: S-BOX GENERATION

The 1STProposed Dynamic S-Box and inverse S-Box generation for odd rounds

The First Step: The s-box is generated by selecting two shared bytes from the sender and recipient and converting them to a binary representation of these bytes. The binary representation is then subjected to an XOR with two adjacent bits until 15 bits are obtained and stored in a 16 × 16 blank matrix. Then, bit number 16 is obtained by XOR bit number 16 with the first bit of the new array, and bit number 17 is obtained by XOR with the first and second bit of the new array, and so on until 256 bits are obtained as illustrated in Figure 2, for example, the two bytes 'rz' can be presented in binary as '0111001001111010. The primary bytes in the banner representation are ignored to generate fresh starting values, even if the generation method is announced and even if the s-box itself is known, enhancing the algorithm's security and making it difficult to predict the s-box values. The binary representation is converted to bytes, yielding a 2 × 16-byte (32-byte) matrix.

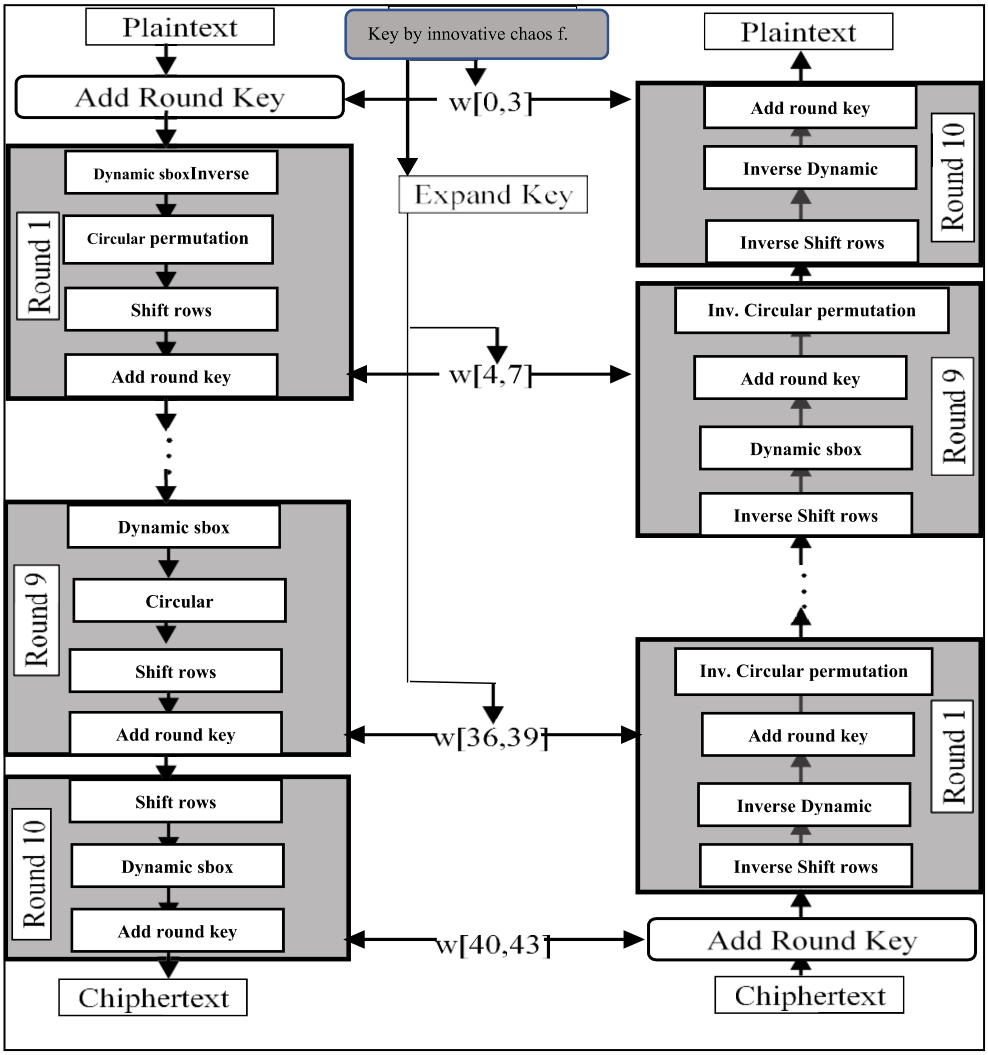


FIGURE 1. Proposed enhanced AES structure.

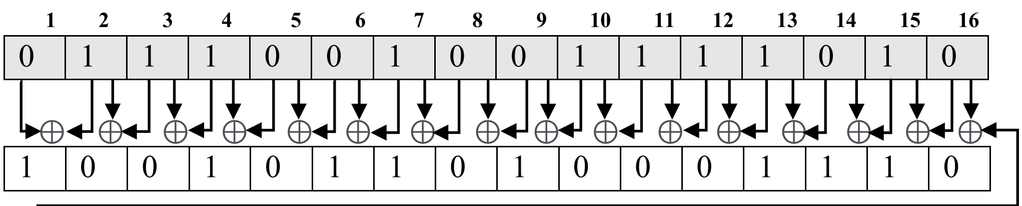


FIGURE 2. First step to S-box generation.

The Second Step: Another 2 × 16 bytes (32) are obtained by shifting to the right. The shifting is made by the values of the second digit after the decimal point of the value f(x) defined in Equation (1); to avoid revealing the main values that were used to generate keys using 16 values, where each row is shifted with the corresponding value of these digits as illustrated in Figure 3. This step obtained a new (16 × 2) byte matrix (32 bytes).

The Third Step: The first and second phases are combined to create a matrix (16 × 4) of 64 bytes. A mask is generated on this matrix, where its value helps make equal or nearly equal numbers of ones and zeros if the matrix is converted to binary representation. This step creates a matrix of (16 × 8), or 64 bytes.

The Fourth Step: The matrices that have been obtained from the first, second, and third steps will be merged, and a

0.6166666666666671 The highlighted value is the shift value of the 1st row of the matrix 0.0827777777778184 The highlighted value is the shift value of the 2nd row of the matrix
0.37357407407412513 The highlighted value is the shift value of the 3rd row of the matrix

0.027914197531504925

FIGURE 3. **The process of shifting the values using the second step of the proposed algorithm.**

0.6166666666666671 The highlighted value is the shift value of the 1st row of the matrix 0.0827777777778184 The highlighted value is the shift value of the 2nd row of the matrix
0.37357407407412513 The highlighted value is the shift value of the 3rd row of the matrix

0.027914197531504925

FIGURE 4. **The process of shifting the values using the fourth step of the proposed algorithm.**

(16 × 8) byte matrix is obtained. A shift is made to the left by the values in third digit after the decimal point as illustrated in Figure 4 resulting in a new (16 × 8) matrix.

The Fifth Step: In this step the matrices from four phases will be used to create a 256-byte matrix. This matrix is subjected to simple equations to enhance complexity and security. Even if the two bytes in the first step are exposed, the s-box remains difficult to identify because creating the s-box is related to knowing the key.

$$x = 255 - expanded\ key, \tag{5}$$

$$y = 255 - final\ step[16][16], \tag{6}$$

$$sbox = (x + y)\%256. \tag{7}$$

where x, and y represent the stored key value and the final s-box stored step, respectively.

The SHA-256 hash method expands a key if its size is less than 256 bytes, eliminating duplicates and adding remaining integers between 0 and 255. This generation starts with two bytes, expands them, and then forms an S-box. Note (Appendix A illustrates the 1st S-box generation and its inverse).

The 2nd proposed S-Box and S-Box Inverse for even rounds generated by this equation:

After deleting any duplicates, the remaining integers are added in the range of [0, 255].

$$sbox[row][column] = (3 \times (row \times 16 + column) + j)\%255 \tag{8}$$

The 1st and 2nd S-Boxes Inverse Generation:

TABLE 2. **Time Comparative with standard AES.**
Plain Text = {'00' '11' '22' '33' '44' '55' '66' '77' '88' '99''aa' 'bb' 'cc' 'dd' 'ee' 'ff'} Cipher Text=]Ù _D Ùå ù/ $û > -ç ãv LK Ï), Þ Ð@ -ž Èñ í p¦

	Standard S-Box	Dynamic S-Box(one sbox)	Dynamic S-Box(two sbox)
Encryption(seconds)	0.20558	0.003562	0.003902
Decryption(seconds)	0.28773	0.001326	0.001382
S-Box Generation (seconds)	1.5555	0.0012474	0.0013842

Each number's row and column values are returned to complete the inverse S-box.

b: REPLACING THE MIX COLUMN WITH CIRCLE PERMUTATION AT THE BIT LEVEL

The time-consuming mix column function has been replaced with an inventive circular permutation function at the bit level due to its speed and high randomness, making it suitable for IoT. It reduces complexity and achieves the concepts of confusion and diffusion; this is accomplished by performing a circular permutation between locations in the 4 × 4 byte text matrix after converting it to its binary representation, which will be 4 × 32 bits in size. The change will include text in the values, not just the locations, after returning it to its byte representation, as shown in Figure 5.

IV. SIMULATION RESULTS AND DISCUSSION

The dimensions of plaintext blocks, key and algorithm sensitivity, and comparative analysis lengths are addressed in this section. Simulations are performed on a Lenovo laptop running Windows 10 (64-bit OS) with Python 3.7, equipped with an Intel(R) CoreTM i5-5200U processor running at 2.20GHz and 8 GB of RAM.

Key generation is based on the parameters of the chaotic map. The key will change drastically with even the smallest alteration. In terms of the algorithm, the s-box plays a crucial role. Two-byte inputs are necessary for the process of generation.

The algorithm and the outcomes are very different because any changes to these two bytes or the chaotic map's parameters will result in a big change. The sbox cryptographic criteria tests (Balancedness, Nonlinearity,

Algebraic Complexity, Strict Avalanche Criterion, differential cryptanalysis) were tested for both S-boxes for even and odd cycles.

Our study uses a chaotic map to generate a primary with a length based on algorithm setting. The text is divided into blocks of 16 bytes, with block padding applied if the final text block is smaller than the permitted size.

Tests were acceptable showing that the two boxes enjoy high efficiency, safety and that the s-boxes implementation

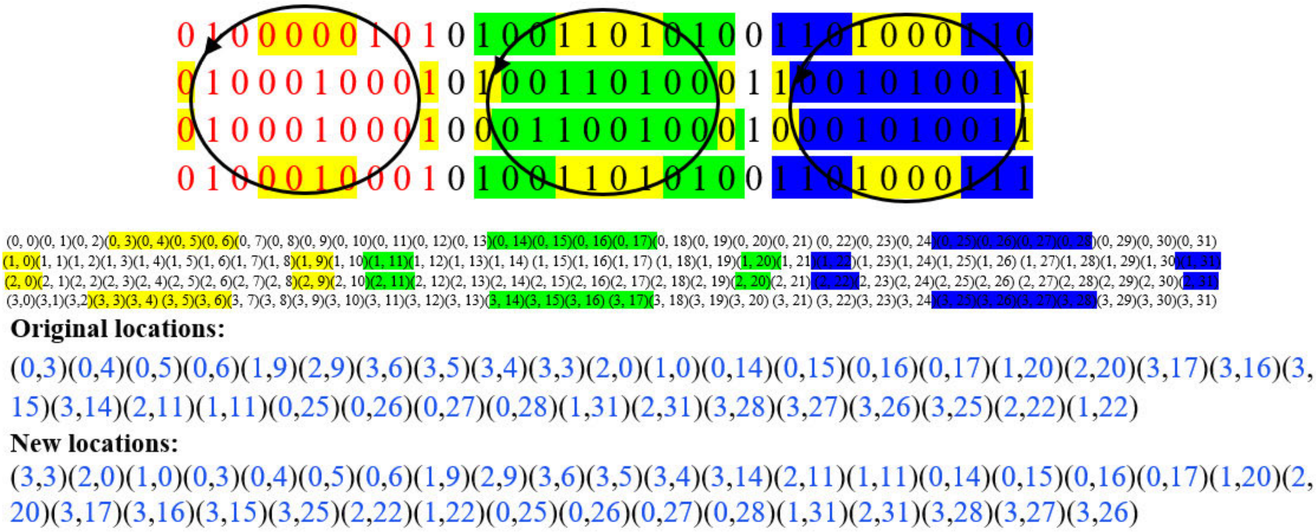
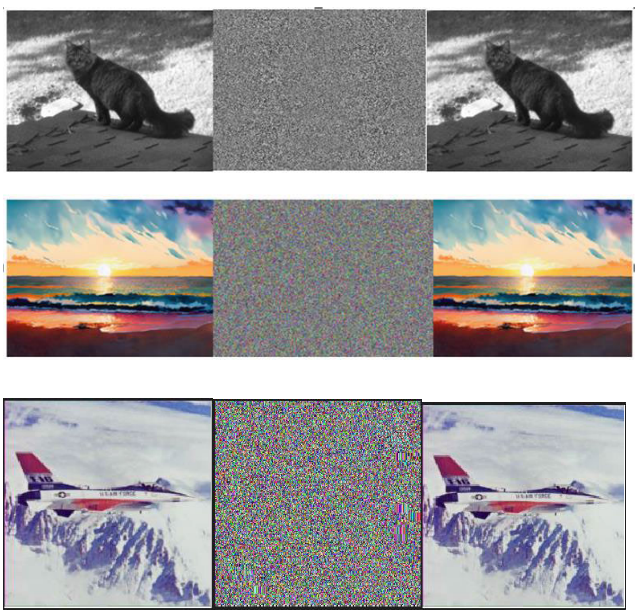


FIGURE 5. The permutation process using the proposed algorithm.



(a) Original (b) Encrypted image (c) Decrypted image

FIGURE 6. The encryption and decryption images using our proposed algorithm.

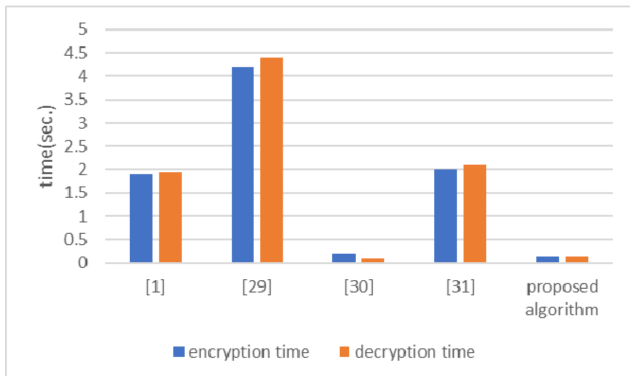


FIGURE 7. Time Comparison (KB/s) with Other Algorithms(taken from [1], [29], [30], and [31]).

time is short, only takes one millisecond. Also, the encryption and decryption time of the algorithm is considered ideal. Our proposed algorithm is applied to encrypt images for security purposed while transmitted over wireless communication channels. Figure 6 shows the results of the encrypted and decrypted images.

The effectiveness of our algorithm is evaluated by assessing the CPU time for encryption, decryption, key setup, and S-Box creation, comparing standard methods and enhanced AES with two dynamic S-Boxes, and replacing the mix column with circular permutation at the bit level. Table 2 shows the encryption and decryption process time for plain text and its corresponding cipher text, which clearly indicated that our proposed technique shows improved time measure. Figure 6 shows our proposed techniques benchmarked with the literature.

Table 3 shows various quality measures of NIST test for our proposed technique the NIST data test is a standardized evaluation process used to appreciate the performance of different algorithms, especially those connected to data handling, analysis, and security.

V. CONCLUSION

The proposed work results in enhancing the AES algorithm for IoT devices by making it lightweight due to the rapid and stochastic nature of these devices, utilizing two key generation methods: an innovative chaotic and Lorenzo's 3D chaotic maps; using two separate dynamism boxes, for even and odd

TABLE 3. NIST test outcome.

Test Name	[17]	[32]	[33]	Propose algorithm
Frequency	0.511369	0.873	0.670	0.534146
Block Frequency	0.515017	0.598	0.485	0.739918
Cumulative Sums	0.528055	0.844	0.135	0.911413
Runs	0.544917	0.758	0.330	0.122325
Longest Run	1.000	0.639	0.100	0.213309
Binary Matrix Rank	0.000000	0.576	0.677	0.911413
Fast Fourier Transform	0.381350	0.889	0.618	0.739918
Non Overlapping Template	1.000000	0.843	0.389	0.911413
Overlapping Template	1.000000	0.793	0.203	0.066882
Universal	0.600640	0.844	0.511	0.000000
Approximate Entropy	1.000000	0.571	0.483	0.534146
Random Excursions	0.941084	0.931	0.594	N/A
RandomExcursions Variant	0.983947	0.502	0.573	N/A
Serial	0.498961	0.861	0.373	0.350485
Linear Complexity	1.000000	0.766	0.462	0.739918

rounds. With high dynamism, its generation time does not take more than 1.2 milliseconds. Changing one of the two bytes that have been used in generating the s-box or one of the parameters of the chaotic map will lead to changing the output of the s-box. The mix column function was replaced by an inventive function that involves cyclic permutations at the bit level. The updated approach offers flexibility and speed in image retrieval, enhancing performance and efficiency in IoT data encryption, particularly for color and grayscale images, with an identical level of precision.

Moreover, NIST testing of our method revealed that it complies with the accepted encryption requirements. The updated AES algorithm, which complies with encryption requirements, has the potential to ensure safe encryption of IoT data. The key generation approach, particularly in IoT applications, offers security and efficiency and meets encryption criteria, demonstrating how cryptographic algorithms can be improved to meet evolving cybersecurity needs.

Future works will involve the use of the proposed techniques for the encryption and decryption of video data. Another direction for research will involve proposing new lightweight appropriate for IoT devices have the same properties as this method.

APPENDIX A

Example of 1st S-Box Generation for Odd Rounds:

First Step: choose two bytes = 'rz' and convert them to it is binary representation:

[0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0]

The process involves performing XOR for each two-bit adjacent bit, as explained in the first step.

[1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0]
[1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1]
[1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0]
[0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0]
[1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1]
[0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1]
[0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1]
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1]
[0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]
[0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1]
[1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0]
[1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1]
[0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1]

It is important to note that the two bytes used for the beginning are not inserted into the S-box values; only the new values are obtained from them. In this step we get 16*2(32 byte).

Second Step: The shift to the right is adjusted by the number of values produced by the new chaotic function used to create the key, with values coming in second after the comma.

Shift values: [1,8,7,2,8,3,2,6,5,3,7,3,4,6,6,2]

The shift list represents the shift amount for each row in the matrix, resulting from the first step.

[0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1]
[1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1]
[0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1]
[0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1]
[0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0]
[1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1]
[0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0]
[0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1]
[1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1]
[1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1]
[0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0]
[0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0]
[1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0]
[1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1]
[1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0]

In this step, we get 16*2 bytes (32 bytes).

Third Step: The two matrices that come from the first and second steps should be combined to create a matrix that is 16 * 4 bytes. Apply a mask to the new matrix.

Mask matrix = [[3, 10, 9, 7], [4, 6, 5, 4], [2, 8, 14, 1], [13, 15, 11, 12]]

The resulting array will be new 16 * 4 bytes

193	177	51	241
221	77	133	192
151	111	1	249
249	27	140	108
60	84	87	211
213	41	100	166
46	184	86	6
164	2	47	80
81	189	92	48
117	151	66	253
37	37	148	218
34	202	117	72
56	234	230	218
214	108	11	133
169	127	142	201
254	222	84	199

Fourth Step: To create a 16 * 8-byte matrix, merge the matrices from the first, second, and third steps. Shift the matrix to the left by the new chaotic function values in the third place after a comma.

Shift values: [2, 6,2,3,7,4,0,6,2,2,6,1,1,0,0,6]

The shift list represents the shift amount for each row in the matrix, resulting from the first step.

165	163	210	209	193	177	51	241
238	78	78	238	221	77	133	192
102	165	75	204	151	111	1	249
170	110	138	187	249	27	140	108
239	86	86	239	60	84	87	211
2	175	224	85	213	41	100	166
193	124	80	63	46	184	86	6
32	76	49	128	164	2	47	80
96	212	162	7	81	189	92	48
10	215	121	194	117	151	66	253
240	195	133	227	37	37	148	218
19	68	162	72	34	202	117	72
155	103	121	182	56	234	230	218
173	168	162	182	214	108	11	133
189	126	249	246	169	127	142	201
108	44	27	11	254	222	84	199

Fifth Step: To create a 16 * 16-byte matrix, combine matrices from the first, second, third, and fourth phases, then implement simple equations using a 256-byte array.

Key=(b'\x8d\xc2_\x88\x14\xc83\xd8\xe5DF\xc7\x7f\xceWp')

Key expansion by the SHA-256 hash algorithm.

x = 255 – expanded key

y = 255 – final step [16][16]

s-box = (x + y) % 256

We obtain this s-box after removing duplicate numbers and adding the remaining numbers between [0, 255].

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

Dynamic S-box

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

Inverse Dynamic S-box

REFERENCES

[1] S. A. Jassim and A. K. Farhan, "Designing a new lightweight AES algorithm to improve the security of the IoT environment," *IRAQI J. Comput. Commun. Control Syst. Eng.*, vol. 22, no. 2, pp. 96–108, 2022.

[2] S. Y. Tarabay, A. Twakol, A. S. Samrah, and I. Yasser, "A secure and efficient cryptography system based on chaotic maps for securing data image in fog computing," *Int. J. Comput. Netw. Inf. Secur.*, vol. 15, no. 1, pp. 64–80, Feb. 2023, doi: 10.5815/ijcnis.2023.01.06.

[3] A. Ahmed, M. M. Madboly, and S. K. Guirguis, "Securing signal encryption based on reduced round homomorphic AES," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 3, 2023, doi: 10.22266/ijies2023.0630.35.

[4] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, "A lightweight AES algorithm implementation for secure IoT environment," *Iraqi J. Sci.*, vol. 1, pp. 2759–2770, Aug. 2021.

[5] M. Abd Zaid and S. Hassan, "Proposal framework to light weight cryptography primitives," *Eng. Technol. J.*, vol. 40, no. 4, pp. 516–526, Apr. 2022.

[6] F. T. A. Hussien, A. M. S. Rahma, and H. B. A. Wahab, "A secure environment using a new lightweight AES encryption algorithm for e-commerce websites," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, Dec. 2021.

[7] A. T. Maolood, E. K. Gbashi, and E. S. Mahmood, "Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 5, p. 4988, Oct. 2022, doi: 10.11591/ijece.v12i5.pp4988-5000.

[8] J. Rokan Naif, G. H. Abdul-majeed, and A. K. Farhan, "Internet of Things security using new chaotic system and lightweight AES," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 11, no. 2, pp. 45–52, Sep. 2019.

[9] A. K. Farhan and E. K. Gbashi, "Chaotic system and DNA computing operations for image encryption based on pixels shuffling," *Al-Qadisiyah J. Pure Sci.*, vol. 26, no. 2, pp. 1–14, Apr. 2021, doi: 10.29350/qjps.2021.26.2.1265.

[10] D. S. Jat and I. S. Gill, "Enhanced advanced encryption standard with randomised round keys," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability*, Jul. 2020, pp. 381–386.

[11] Y. Hussain Ail and Z. A. H. Alobaidy, "Images encryption using chaos and random generation," *Eng. Technol. J.*, vol. 34, no. 1, pp. 172–179, Jan. 2016.

[12] A. Gupta and A. Gupta, "A new technique of image encryption using modified AES algorithm," *Int. J. Multidiscip. Innov. Res.*, vol. 1, pp. 228–2583, 2021.

[13] V. Ashqi Saeed and B. Haval Sadiq, "Image encryption based on AES algorithm and XOR operation," *Academic J. Nawroz Univ.*, vol. 12, no. 3, pp. 533–539, Aug. 2023.

[14] R. B. Antonio, A. M. Sison, and R. P. Medina, "A modified generation of S-box for advanced encryption standards," in *Proc. 2nd Int. Conf. Inf. Sci. Syst.*, Mar. 2019, pp. 280–283.

[15] A. Nitaj, W. Susilo, and J. Tonien, "A new improved AES S-box with enhanced properties," in *Proc. Inf. Secur. Privacy, 25th Australas. Conf.*, 2020, pp. 125–141.

[16] F. J. D'souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 647–652.

[17] A. Yousif and A. H. Kashmar, "Key generator to encryption images based on chaotic maps," *Iraqi J. Sci.*, vol. 1, pp. 362–370, Feb. 2019.

[18] D. S. Jat and I. S. Gill, "Enhanced advanced encryption standard with randomised S box," in *Proc. 5th Int. Conf. Innov. Technol. Intell. Syst. Ind. Appl. (CITISIA)*, Nov. 2020, pp. 1–6.

[19] N. Ibrahim and J. Agbinya, "Design of a lightweight cryptographic scheme for resource-constrained Internet of Things devices," *Appl. Sci.*, vol. 13, no. 7, p. 4398, Mar. 2023.

[20] A. Kadhim and R. S. Ali, "Enhancement AES based on 3D chaos theory and DNA operations addition," *Karbala Int. J. Modern Sci.*, vol. 5, no. 2, pp. 1–23, Jul. 2019.

[21] A. T. Khudhair, E. K. Gbashi, and A. T. Maolood, "Novel dynamic S-box based on password key and circle map," *Iraqi J. Sci.*, vol. 1, pp. 4767–4778, Sep. 2023.

[22] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, p. 1484, Dec. 2019.

[23] R. Saha, G. Geetha, G. Kumar, and T.-H. Kim, "RK-AES: An improved version of AES using a new key generation process with random keys," *Secur. Commun. Netw.*, vol. 2018, pp. 1–11, Nov. 2018.

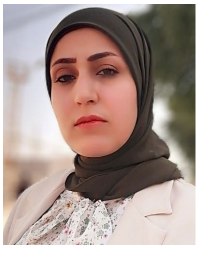
[24] A. M. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptogr. Netw. Secur.*, vol. 16, no. 1, p. 11, Sep. 2017.

[25] N. Dalakoti, N. Gaur, and A. Mehra, "Hardware efficient AES for image processing with high throughput," in *Proc. 1st Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Sep. 2015, pp. 932–935.

[26] A. Moneem S. Rahma and M. Abid Ali K, "To modify the partial audio cryptography for Haar wavelet transform by using AES algorithm," *Eng. Technol. J.*, vol. 32, no. 1B, pp. 169–182, Jan. 2014.

[27] K. Shahbazi and S. Ko, "High throughput and area-efficient FPGA implementation of AES for high-traffic applications," *IET Comput. Digit. Techn.*, vol. 14, no. 6, pp. 344–352, Nov. 2020.

[28] L. M. Jawad, "A new scan pattern method for color image encryption based on 3D-lorenzo chaotic map method," *Multimedia Tools Appl.*, vol. 80, no. 24, pp. 33297–33312, Oct. 2021.
[29] G. Manjula and H. S. Mohan, "Improved dynamic S-box generation using hash function for AES and its performance analysis," in *Proc. 2nd Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Aug. 2018, pp. 109–115.
[30] L. Sleem and R. Couturier, "Speck-R: An ultra light-weight cryptographic scheme for Internet of Things," *Multimedia Tools Appl.*, vol. 80, no. 11, pp. 17067–17102, May 2021, doi: 10.1007/s11042-020-09625-8.
[31] S. Niveda, A. S. Sakthi, S. Srinitha, V. Kiruthika, and R. Shanmugapriya, "A novel Simon light weight block cipher implementation in FPGA," in *Pervasive Computing and Social Networking*. Cham, Switzerland: Springer, 2022, pp. 159–170.
[32] R. S. Mohammed, "Design a lightweight authentication encryption based on stream cipher and chaotic maps with sponge structure for Internet of Things applications," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 1, pp. 1–16, 2023.
[33] J. Daemen, S. Hoffert, M. Peeters, G. Van Assche, and R. Van Keer, "Xoodyak, a lightweight cryptographic scheme," *IACR Trans. Symmetric Cryptol.*, vol. 1, pp. 60–87, Jun. 2020.



A'LAA HUSSEIN ALI received the bachelor's degree in computer sciences data security branch from the University of Technology (UOT), Baghdad, Iraq, in 2014, where she is currently pursuing the degree in computer science, specializing in data security. She has been a Programmer with the Health Ministry of Iraq, since 2019.

EKHLAS KHALAF GBASHI received the bachelor's and master's degrees in computer sciences from the University of Technology (UOT), Baghdad, Iraq, in 1998 and 2005, respectively, and the Ph.D. degree in networks security from the Department of Computer Sciences, Technology University. She has been a Faculty Member with the Computer Sciences Department, UOT, since 2000, where she was the Head of Computer Security Branch, UOT, from 2016 to 2020. Her research interests include networks security (intrusion detection systems), data security, computer networks, comparative education and computer architecture, image processing, and artificial intelligence (AI).

HAYA ALASKAR received the M.Sc. degree in applied artificial intelligence from the University of Exeter, in 2009, and the Ph.D. degree in computer science from Liverpool John Moores University, in 2014. She is currently an Assistant Professor with the College of Computer Science and Engineering, Prince Sattam Bin Abdulaziz University, Saudi Arabia. She has several publications concentrated on using machine learning in various medical data, such as signals and images. Her research interests include artificial intelligence applications and data science.

ABIR JAAFAR HUSSAIN (Senior Member, IEEE) received the Ph.D. degree from The University of Manchester (UMIST), U.K., in 2000. Her Ph.D. thesis titled Polynomial Neural Networks for Image and Signal Processing. She is currently a Professor of image and signal processing with the University of Sharjah, United Arab Emirates, and also a Visiting Professor with Liverpool John Moores University, U.K. She is the Ph.D. Supervisor and an External Examiner of research degrees, including the Ph.D., and M.Phil. She has published numerous referred research papers in conferences and journals in the research areas of neural networks, signal prediction, telecommunication fraud detection, and image compression. She has worked with higher order and recurrent neural networks and their applications to e-health and medical image compression techniques. She has developed with her research students a number of recurrent neural network architectures. She is one of the initiators and chairs of the Development in e-Systems Engineering (DeSE) conference series.