Tailoring AES for resource-constrained IoT devices

Shaimaa S. Saleh[1], Amr A. Al-Awamry[1], Ahmed Taha[2]
[1]Department of Electrical Engineering, Benha Faculty of Engineering, Benha University, Benha, Egypt
[2]Department of Computer Science, Faculty of Computer and Artificial Intelligence, Benha University, Benha, Egypt

Article Info

Article history:

Received Jan 22, 2024
Revised May 27, 2024
Accepted Jun 5, 2024

Keywords:

AES algorithm
Cryptography
ESP32
IoT security
Substitution-box

ABSTRACT

The internet of things (IoT) is a network of interconnected hardware, software, and many infrastructures that require cryptography solutions to provide security. IoT security is a critical concern, and it can be settled by using cryptographic algorithms such as advanced encryption standard (AES) for encryption and authentication. A fundamental component within the AES algorithm is the substitution box (S-box), which generates confusion and nonlinearity between plaintext and ciphertext, strengthening the process of security. This paper introduces a comparative analysis to offer valuable knowledge of the factors related to different S-box modifications, which will ultimately affect the design of cryptographic systems that use the AES algorithm. Then, a tailored AES algorithm is proposed for resource-constrained IoT devices by changing the standard S-box with another S-box. The new S-box reduces the rounds number and the time needed for the AES algorithm's encryption, decryption, and key expansion. The performance of the proposed AES is assessed through various experiments. Therefore, our tailored AES with the new S-box is more secure and efficient than AES with a standard S-box.

Corresponding Author:

Shaimaa S. Saleh
Department of Electrical Engineering, Benha Faculty of Engineering, Benha University
Benha, Egypt
Email: shimaa.said@bhit.bu.edu.eg

1. INTRODUCTION

Internet of things (IoT) has been widely employed globally for various purposes such as minimizing human efforts, achieving effectiveness, and fluently understanding client needs [1]. IoT is being used in healthcare technology, manufacturing, banking, retail, logistics, and different industries, but unfortunately, they face serious security attacks. Assaulters, including unauthorized individuals, can gain access to IoT devices and exploit the network for malicious purposes or even the device itself. security becomes more critical. When these devices are designed for human security or health services [2]. Major requirements must be satisfied to guarantee IoT security: confidentiality means securing data from illegal exposure, safety means data cannot be changed if previous authorization has not been attained, and availability ensures access to data on demand.

Cryptography algorithms are employed to guarantee confidentiality, while message authentication code (MAC) devices and digital signatures are utilized to ensure security [3], [4]. Cryptography is a potent technology that protects data and transactions within the IoT. The advanced encryption standard (AES) is widely adopted among encryption methods. AES is a commonly used symmetric essential encryption technique that is considered a cornerstone of modern cryptographic systems. Its importance stems from its strong security and the careful engineering of its constituent parts, of which the substitution-permutation network (SPN) is a key component. The substitution box (S-box) also called the S-box, adds nonlinearity and confusion to the SPN and strengthens AES's defense against several cryptographic assaults [5]. There are many S-box generation methods using different techniques like deoxyribonucleic acid (DNA) computing [6], [7], elliptic curves [8], [9], linear fractional transformation [10], [11] and compressive sensing [12], [13]. The S-box strength can be measured using a variety of criteria given by National Institute of Standards and Technology (NIST), including bit independence criterion (BIC), nonlinearity, differential uniformity, bijectivity, and the strict avalanche criterion (SAC) [14]. A cursory review of the literature indicates that the cryptography community is constantly working to improve AES's security features, with the S-box being a primary area of interest [6], [15]-[27].

The AES algorithm is employed as one of the methods to ensure the privacy and confidentiality of transferred data over different computer networks. Additionally, it is globally recognized as one of the most extensively adopted symmetric block cipher algorithms. It boasts a distinctive structure for both encrypting and decrypting valuable data. It can be implemented effectively in either hardware or software environments. AES is a symmetric key block cipher with 128-bit [5]. It is offered in three distinct variants: AES-128, AES-192, and AES-256, with key lengths of 128 bits, 192 bits, and 256 bits, respectively. The AES algorithm operates on 128-bit intermediate results, organized in a 4×4 matrix known as the state. Each element within the matrix represents a byte, with indices ranging from 0 to 15. The matrix rows follow the pattern (j, j+4, j+8, j+12), and the columns are structured as (4j, 4j+1, 4j+2, 4j+3) for $0 \leq j \leq 3$. The algorithm is composed of multiple rounds, and each round (except the final round) consists of four transformations: SubBytes, ShiftRows, MixColumns, and add round key. The number of rounds varies depending on the AES version: AES-128 employs ten rounds, AES-192 uses twelve rounds, and AES-256 consists of fourteen rounds. Importantly, the last round in each version excludes the MixColumns operation [28]. The AES standard adheres to Rijndael's specifications for key and block sizes. The four primary transformations in the AES algorithm are described as follows:

i) The substitution bytes (SubBytes) transformation substitutes each byte in the input state array with another byte retrieved from a nonlinear substitution table (S-Box).
ii) Shifting of rows: this transformation promotes diffusion within the columns of the input state matrix [5].
iii) Mixing of columns: it involves multiplying the columns of the state array by a fixed matrix and the multiplication results are computed over the galois field (GF) (2^8).
iv) Add round key: in this step, 128 state bits undergo bitwise XOR with the key specific to that round. The add round key operation is performed one column at a time.

Although AES is secure, there is still room for improvement. This paper introduced a comparative evaluation of enhanced AES S-box designs to enhance our comprehension of how these improvements influence the AES algorithm's overall security and cryptographic strength, then introduced a new version of the AES 128 that uses a new S-box with fewer iteration rounds. The objective of this modified AES 128 is to minimize the encryption, decryption, and key expansion time for the plain text and decrease the memory size used while preserving a suitable level of security which is measured in terms of the avalanche effect. The ESP32 card is chosen as the platform for implementing AES-128 encryption due to its numerous advantages and specific design catered towards supporting Internet of Things projects.

2. REVIEW OF RELATED LITERATURE

Security is a significant challenge in IoT. Numerous researchers have been working on addressing security issues, and the following sections highlight their efforts and contributions. Chowdhury *et al.* [29] introduces a modified advanced encryption standard (MAES), which is a simple version of the AES designed to meet specific requirements. It introduces a novel equation to construct a square matrix in the MAES affine transformation stage, resulting in a new 1-dimensional S-box. MAES boasts an approximately 18.35% packet transmission efficiency rate, making it more energy-efficient than AES. This makes MAES a suitable choice for environments with limited resources.

Hassan and Zaid [30], a modified algorithm for AES aimed at addressing lightweight constraints was introduced. In this proposal, the mixed-column process and the round key's addition are combined into a single cycle. Additionally, they modified the shift row process to encompass both row and column shifting. As a result, the rounds number was decreased to six, significantly streamlining the algorithm. This modified algorithm successfully passed the statistical tests and exhibited superior speed compared to the standard AES. Moreover, it is well-suited for deployment in optimized resource-constrained environments as in devices of IoT.

The study in [31] employs a 3-dimensional S-box to illustrate a significant scheduling technique known as the S-box. Security enhancements are achieved by implementing the logistic map method. This proposed method is well-suited for lightweight IoT devices, such as smart home appliances. That research investigates the impact of introducing chaos on expediting the planned initialization of keys prior to message transmission. The evaluation of the proposed technique involves assessing the generation delay needed for smart home-sensor devices.

Fadhil *et al.* [32] introduces modifications in the standard AES to create a lightweight version called LAES, depending on chaotic systems. The LAES algorithm employs functions similar to the conventional AES, but with certain alterations in the form of substituting ShiftRows with initial permutation (IP) and employing dynamic ShiftRows instead of MixColumns, reducing processing time. Encryption tools were developed utilizing three chaotic maps. The process involved three phases. In the first phase, a S-box was generated using a 1-dimensional chaos map. The output of this phase served as the initial variables for the second phase's 2-dimensional chaos map, which generated two IPs for even and odd rounds. In the final phase, secret keys and shifting value arrays were created by importing a 3-dimensional chaos map, which was created from the 2-dimensional chaos map that was the output of the second phase. There were three types of chaotic keys created: one for odd rounds, another one for even rounds, and last one for the starting round. The research findings showed that LAES was capable of encrypting a text file which size 5 MB under 1.987 seconds.

Moreover, Alshammari *et al.* [33] introduced a lightweight encryption technique based on AES. They initiated the process by utilizing chaotic Boolean functions to construct a secure cryptographic S-box. The diffusion and permutation steps were implemented using the Lorenz system and Hilbert curve scan pattern, respectively. The features of the chaotic S-cryptographic box were evaluated along with NIST testing to assess its strength. The method was specifically designed to address the constraints of highly limited IoT devices. Experimental data confirmed that the method is lightweight enough to fulfill various requirements, involving low memory consumption, fast execution time, and sufficient information entropy.

Rahman *et al.* [34] presented an encryption system that enhanced the AES algorithm by incorporating a white box and employing a doubled encryption approach. Instead of using the standard substitute-byte S-box in AES, this method utilized a white box. The introduction of a white box was crucial as it segmented the whole AES encryption into several round functions. Repeating the AES encryption process added resilience against potential disruptions caused by attackers or malware, thereby increasing the difficulty of compromising the network or system. The proposed solutions were found to be effective in preventing denial of service (DoS) attacks on small devices and other internet of thing's devices, according to the study's findings.

Hammod [35] introduces modifications to the AES technique by altering the SubByte and the ShiftRows stage in the encryption section and the InvSubByte and InvShiftRows stage in both encryption and decryption sections. These modifications were made to adapt the classical AES method with the proposed methodology. Evaluation of the results revealed that ML-AES demonstrated better outcomes in terms of enhanced transmission criteria. Additionally, ML-AES demonstrated a 10% improvement in speed compared to the original AES, while also consuming less memory. Furthermore, the computed avalanche for diffusion properties in ML-AES outperformed the classic AES algorithm.

3. SYSTEM BACKGROUND

This section introduces a comparative analysis between a standard and a new S-box. The variant criteria given by NIST offer valuable knowledge of the factors related to different S-box modifications. This will ultimately affect the design of the AES algorithm's cryptographic systems.

3.1. AES S-box structure

The S-box of AES consists of a matrix containing 256 elements, and the values within this matrix range from 0 to 255. The values of S-box are calculated with $S = ny \oplus C \ mod \ M$ where y is the input byte, n is 8×8 affine matrix, and C is affine constant $63_{16} = 01100011_2$ and M is an irreducible polynomial $y^8 + y^4 + y^3 + y + 1$. Every element in the S-box is paired with its multiplicative inverse in GF (2^8) with the exception of element 0, which is mapped to itself. During the process of decryption, AES utilizes the inverse S-box [36].

The AES S-box is formed by combining two transformations, denoted as f and g. For $y \in GF\ (2^8)$ by $S(y) = g \circ f(y) \ where$ $\circ$ denotes the polynomial multiplication,

− The nonlinear function f defined by:

$$f(y) = \begin{cases} y^{-1} & y \neq 0 \\ 0 & y = 0 \end{cases} \tag{1}$$

− The affine function defined by $g(y) = ny + C$ for a field element:

$$y = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)\ , s = ny + c$$

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{2}$$

3.2. New S-box structure

The new S-box is exceptionally non-linear based on highly non-linear bent functions, which are expanded by a biological technique depending on DNA. This S-box functions as a one-to-one transformation, replacing a byte with its corresponding value. Significantly, it is a reversible function obtained through a sequence of transformations.

- The affine function denoted by:

$$W = T(aX^2 + bX + C)$$

$$= \begin{bmatrix} a_4\, a_3\, a_2\, a_1\, a_0\, a_7\, a_6\, a_5 \\ a_5\, a_4\, a_3\, a_2\, a_1\, a_0\, a_7\, a_6 \\ a_6\, a_5\, a_4\, a_3\, a_2\, a_1\, a_0\, a_7 \\ a_7\, a_6\, a_5\, a_4\, a_3\, a_2\, a_1\, a_0 \\ a_0\, a_7\, a_6\, a_5\, a_4\, a_3\, a_2\, a_1 \\ a_1\, a_0\, a_7\, a_6\, a_5\, a_4\, a_3\, a_2 \\ a_2\, a_1\, a_0\, a_7\, a_6\, a_5\, a_4\, a_3 \\ a_3\, a_2\, a_1\, a_0\, a_7\, a_6\, a_5\, a_4 \end{bmatrix} \times \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix}^2 + \begin{bmatrix} b_4\, b_3\, b_2\, b_1\, b_0\, b_7\, b_6\, b_5 \\ b_5\, b_4\, b_3\, b_2\, b_1\, b_0\, b_7\, b_6 \\ b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0\, b_7 \\ b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0 \\ b_0\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1 \\ b_1\, b_0\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2 \\ b_2\, b_1\, b_0\, b_7\, b_6\, b_5\, b_4\, b_3 \\ b_3\, b_2\, b_1\, b_0\, b_7\, b_6\, b_5\, b_4 \end{bmatrix} \times \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} + \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix} \tag{3}$$

$$a = 0x76, \quad b = 0x6D,\ C = 0xDA$$

- The computation of the multiplicative inverse is done using W: $W = W^{-1}GF(2^8)$, which can be denoted by:

$$W = W^{-1} = \begin{cases} W^{254} & W \neq 0 \\ 0 & W = 0 \end{cases} \tag{4}$$

- There affine function is applying again as follow.

$$W = T(aW^2 + bW + C)$$

$$= \begin{bmatrix} a_4\, a_3\, a_2\, a_1\, a_0\, a_7\, a_6\, a_5 \\ a_5\, a_4\, a_3\, a_2\, a_1\, a_0\, a_7\, a_6 \\ a_6\, a_5\, a_4\, a_3\, a_2\, a_1\, a_0\, a_7 \\ a_7\, a_6\, a_5\, a_4\, a_3\, a_2\, a_1\, a_0 \\ a_0\, a_7\, a_6\, a_5\, a_4\, a_3\, a_2\, a_1 \\ a_1\, a_0\, a_7\, a_6\, a_5\, a_4\, a_3\, a_2 \\ a_2\, a_1\, a_0\, a_7\, a_6\, a_5\, a_4\, a_3 \\ a_3\, a_2\, a_1\, a_0\, a_7\, a_6\, a_5\, a_4 \end{bmatrix} \times \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \end{bmatrix}^2 + \begin{bmatrix} b_4\, b_3\, b_2\, b_1\, b_0\, b_7\, b_6\, b_5 \\ b_5\, b_4\, b_3\, b_2\, b_1\, b_0\, b_7\, b_6 \\ b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0\, b_7 \\ b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0 \\ b_0\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1 \\ b_1\, b_0\, b_7\, b_6\, b_5\, b_4\, b_3\, b_2 \\ b_2\, b_1\, b_0\, b_7\, b_6\, b_5\, b_4\, b_3 \\ b_3\, b_2\, b_1\, b_0\, b_7\, b_6\, b_5\, b_4 \end{bmatrix} \times \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \end{bmatrix} + \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix} \tag{5}$$

$$a = 0x76, \quad b = 0x6D,\ C = 0xDA$$

The new S-box (in Hex) is represented in Table 1, showing the generated values. These values are converted into binary format, ensuring that the length is a multiple of eight. If not, the number is adjusted by adding zeroes to the left. Afterward, each pair of bits is converted into a DNA code [6], and the inverse new S-box (in Hex) is represented in Table 2.

Table 1. The new S-box (Hex)

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

Table 2. The inverse new S-box (Hex)

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

3.3. Performance analysis of S-box

The structure of S-box determines block ciphers' strong performance to create a secure cryptosystem that can withstand attacks. The AES system's S-box first layer needs to satisfy several requirements, which will be illustrated in this section. A detailed performance analysis of the new s-box compared to the standard s-box will also be discussed below. The S-box was built using the MATLAB program, and its performance was analyzed.

3.3.1. Strict avalanche criterion

Strict avalanche criterion (SAC) quantifies the impact of altering a single input bit on the corresponding output bits within a cryptographic system. For a cryptographic S-box, a change in one input bit would, on average, cause about half of the output bits to be changed [6]. This characteristic adds to the diffusion and confusion elements within the cryptographic algorithm. Theorem 1 [6] assumes that $u(x) = (u_1(x),\cdot\cdot\cdot, u_n(x))$ from $GF(2)^n$ to $GF(2)^n$ is a multiple output Boolean function, $\forall B = (B_n, B_{n-1}, \ldots, B_1) \in GF(2)^n$, $w(B) = 1, if\ w(u_i(x + B) + u_i(x)) = 2^{n-1}, (1 \leq i \leq n)$, then $u(x)$ satisfies SAC. We can see from Tables 3 and 4 that the mean value for SAC for the new S-box is 128.125, while it is 129.25 for the AES S-box.

Table 3. The SAC of AES S-box

SAC	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
00000001	132	132	116	144	116	124	116	128
00000010	120	124	144	128	124	116	128	136
00000100	132	132	128	120	144	128	136	128
00001000	136	136	120	116	128	136	128	140
00010000	116	128	116	132	128	128	140	136
00100000	116	132	132	120	120	140	136	136
01000000	136	136	120	132	120	136	136	124
10000000	132	144	132	136	124	136	124	132

Table 4. The SAC of the new S-box

SAC	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\boldsymbol{u_5}$	$\boldsymbol{u_6}$	$\boldsymbol{u_7}$	$\boldsymbol{u_8}$
00000001	128	136	132	128	128	124	128	120
00000010	128	120	124	128	140	128	136	136
00000100	128	136	128	132	136	132	120	116
00001000	132	128	136	136	124	132	136	132
00010000	120	124	128	132	128	112	116	116
00100000	124	124	124	128	128	132	116	128
01000000	124	132	136	128	132	132	140	128
10000000	136	116	132	120	136	132	132	128

3.3.2. Distance to SAC of the new S-box

Theorem 2 assumes that $u(x) = (u_1(x), \cdots, u_n(x))$ from $GF(2)^n$ to $GF(2)^n$ is a multiple output Boolean function, the distance to SAC is symbolled by DSAC (u) and its theorem is:

$$\mathrm{DSAC(u)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \sum_{\substack{\mathrm{B \in GF(2)^n} \\ \mathrm{w(B)=1}}} \left| \mathrm{w}\big(\mathrm{u_i(x+B)+u_i(x)}\big) - \mathrm{2^{n-1}} \right| \tag{6}$$

patently $u(x)$ fulfills SAC when $DSAC(u) = 0$. AES S-box does not fulfill SAC. Table 3 displays the SAC of AES S-box function $u(x) = (u_1(x), \cdots, u_8(x))$. Then, the obtained DSAC values for AES S-box and the new AES S-box are 432 and 316, respectively, as per Theorem 2. Although each output-bit function of the new AES S-box cannot fulfill SAC, the amount of modified output bits changed is very close to $2^{n-1} = 128$, that is, the probability approaches 50%. the output bit will inverse with a probability reaching near to 50%, when one bit of the input is inversed.

3.3.3. The algebraic attacks resistance

Theorem 3 RAA is denoted by Γ given h equations of g terms in $GF(2^8)$, and the Γ is defined to be:

$$\Gamma = \left(\frac{(g-h)}{S}\right)^{\left\lceil \frac{(g-h)}{S} \right\rceil} \tag{7}$$

using the AES S-box with $g = 81$, h=23, $S = 8$, we can get $\Gamma = 2^{22.9}$. It was alleged in [37] that Γ should be more than 2^{32} to overcome the disadvantages of the S-box. While AES S-box has $\Gamma = 2^{22.9}$, it can be a weakness of AES. This measure is mostly reliant on the inverse multiplicative. For the new AES S-box, $g = 510$, h=255, $S = 8$, we can obtain $\Gamma = 2^{160}$. This number reflects how resistant the new S-box is to different algebraic attacks.

3.3.4. Bit independence criterion

Webster and Traverses introduced the bit independence criterion (BIC) parameter [38]. It measures the extent to which the output bits are independent of each other. So, it is used to test the security level of the S-Boxes against various attacks. Theorem 4 if $u(x) = (u_1(x), \cdots, u_n(x))$ from $GF(2)^n$ to $GF(2)^n$ is a multiple output Boolean function, then BIC is made by getting $n \times n$ dimensional matrix $BIC(u) = b_{ik}$ such that i, k, and b_{ik} are denoted to be:

$$BIC(u) = \sum_{i=1}^{n} \sum_{\substack{B \in GF(2)^n \\ w(B)=1}} \left| w(u_i(x) + u_k(x) - 2^{n-1}) \right| \tag{8}$$

3.3.5. Non-linearity

Non-linearity (NL) is the most crucial factor in determining how effective S-box are. When the S-box exhibits a linear mapping between plaintext and ciphertext, it is considered weak because linear relationships make it easier for attackers to launch linear attacks. This linear attack can be mitigated by creating S-box with a nonlinear mapping between the plaintext and the ciphertext. The nonlinearity of the n-bit Boolean function can be calculated mathematically using the formula.

Theorem 5 if $u(x) = (u_1(x), \cdots, u_n(x))$ from $GF(2)^n$ to $GF(2)^n$ is a multiple output Boolean function, the nonlinearity that is calculated for n -bit Boolean functions as NL (u_i) is denoted as follows:

$$NL(u_i) = 2^{n-1} - \tfrac{1}{2}\left(\left|W_{u_i}(m)\right|\right) \qquad \text{Where } m \in u_2^n \tag{9}$$

$$W_u(m) = \sum_{t\in\{0,1\}^n} (-1)^{u(t)\oplus t.m} \tag{10}$$

$$NL(u) = \min_{\substack{0\neq v\in GF(2)^n \\ l(x)\in L_n[x]}} d(v.u(x),(x)) \tag{11}$$

where the linear functions from $GF(2)^n$ to $GF(2)^n$ is denoted by $L_n[x]$.

Theorem 5 states that both the AES S-box and the new AES S-box exhibit $NL\ (u)$=112. Corona-Bermúdez *et al.* [16], Cheon and Lee [37] pointed out that the $NL(u)$ of perfect non-linear function should be $NL(u) = 2^{\,n-1} - 2^{\frac{n}{2}-1} = 2^{\,8-1} - 2^{\frac{8}{2}-1} = 120$. Both the AES S-box and the new AES S-box fall short of being perfect nonlinear functions, but NL (AES S-box) and NL (new AES S-box) are very near to the $NL\ (u)$ of ideal nonlinear function.

3.3.6. S-box iterative period

Theorem 6 [39], the S-box bending function is represented by $j\ (n)$. The periodicity is satisfied if $j\ (n)^k = n$ and k is any positive number. Let us take the equation $j\ (n)^k = n\ j\ (n)^k$ for any $n \in GF(2^8)$, the iterative periods found were 2, 27, 59, 81, and 87. These time frames fulfilled 2+27+59+81+87=256, therefore no overlap happens among the period orbits. It is evident that the standard S-box has brief durations and insufficient distribution, that may result in some gap. The new S-box has addressed these issues by increasing the iterative period to its maximum value, reaching 255 for any positive number in the GF(2^8).

3.3.7. Balance criteria and bijectivity

This property represents crucial characteristics for an effective cryptographic S-box. These criteria emphasize the necessity of generating a unique output for every distinct input within the system. In the case of an 8×8 S-box with 8-bit input and output, each distinct 8-bit input must yield an 8-bit unique production. If the S-box lacks bijectivity, it exposes a weak cyclic structure, rendering it susceptible to attacks. Notably, the AES and new S-box adhere to bijectivity and balance, ensuring robust cryptographic performance [26].

3.4. Comparison of different S-boxes

Researchers have investigated several methods to construct S-boxes resistant to cryptographic attacks. A comparison to evaluate the security and effectiveness of several S-boxes arranged from the most recent to the oldest is shown in Table 5. The primary area of emphasis in the designs of S-boxes is the SAC value that quantifies the impact of altering a single input bit on the corresponding output bits. A change in one input bit should, on average, cause about 0.5000 of the output bits to change. Nonlinearity is another metric that assesses the link between output and input values; the more significant this metric, the more challenging it will be for attackers to anticipate or reverse its operations. As observed in Table 5, the new S-box [6] has the ideal value of 112. The third feature to compute is the BIC, which measures the degree of statistical independence between the outputs of the S-box when modifications are made to the inputs. The higher will be this value, the more unpredictable and safer the S-box's behavior is thought to be. As a result, the S-box [6] has ideal values compared to other designs.

Table 5. Comparison S-boxes NL, SAC, and BIC

S-box	Year	Max NL	Min NL	Avg NL	Max SAC	Min SAC	Avg SAC	BIC
[15]	2023	110	106	108	0.5938	0.4219	0.4956	103.78
[16] S-box 1	2023	106	100	104	0.5937	0.3906	0.5002	103.92
[16] S-box 2	2023	106	98	104.25	0.609	0.375	0.5029	104
[17]	2023	110	106	107.75	0.5748	0.4217	0.4983	104.14
[18]	2022	108	104	105.25	0.5937	0.4218	0.5070	102.72
[6]	2022	112	112	112	0.53125	0.4375	0.50122	103.406
[19]	2022	110	104	107	0.5781	0.4219	0.4954	102.93
[20]	2021	112	108	110.5	0.5625	0.4219	0.5065	106.43
[8]	2021	108	104	105	0.6400	0.4060	0.5060	103.5
[9]	2021	106	106	106	0.5937	0.4218	0.507	96
[21]	2021	112	110	111.75	0.609	0.374	0.502	103.7
[22]	2020	112	110	111.25	0.5937	0.4062	0.5007	102.57
[23]	2020	110	108	106	0.5781	0.4063	0.4990	104.92
[25]	2020	108	106	106.5	0.5781	0.4219	0.5010	104.07
[26]	2020	108	108	108	0.5983	0.4063	0.4971	103.86
[27] S-box 1,2	2020	104	96	102	0.625	0.375	0.4915	103.57

Table 6 demonstrates, for all cryptographic criteria the performance of the new S-box [6] is equal to or better than the former ones, and it is getting very close to the performance of an optimal S-box. So, the inverse new S-box has been computed and used as a look-up table in our application to get more security and more accurate results for encryption and decryption time in our IoT application. This suggests that the new S-box provides enhanced security compared to its predecessor models.

Table 6. Comparisons of cryptographic properties of S-boxes

Performance index	AES S-box	S-box [40]	S-box [41]	S-box [42]	New S-box [6]	Optimal value
Balance criteria	Balance	Balance	Balance	Balance	Balance	Balance
Differential uniformity (F)	4	4	4	4	4	4
Non-zero linear structure	None	None	None	None	None	None
Algebraic attacks resistance (Γ)	$2^{22.9}$	$2^{22.9}$	$2^{22.9}$	$2^{22.9}$	2^{160}	$> 2^{32}$
Distance to SAC (DSAC)	432	408	372	328	316	0
Nonlinearity (NL)	112	112	112	112	112	120
Number of terms in S-box algebraic expression	9	255	255	255	255	255
Number of terms in inverse S-box algebraic expression	255	9	253	254	255	255
Affine transformation period	4	4	16	254	256	256
Iterative period	Less than 88	less than 88	256	256	256	256

4. THE PROPOSED METHOD FOR THE NEW AES

This section discusses modifications to the AES encryption algorithm, including reducing iteration rounds and substituting standard S-box with a new S-box. Then, the AES 128 algorithm was implemented using the standard S-box and the new S-box on a group of sensors linked to the ESP32 card as a proposed IoT system design to test the strength of the modified AES using the new S-box at four rounds. This modification aims to optimize security and decrease the time of encryption and decryption process.

The new version of AES represents an evolution from the standard AES, incorporating advancements by employing four rounds for encrypting plaintext, operating with a 128-bit block of information, and utilizing a 128-bit key to both decryption and encryption, this algorithm introduces heightened complexity and security measures. Notably, it integrates the new S-box and its inverse, as outlined in Tables 1 and 2, which improves the encryption process's overall security even more. The new algorithm for AES uses four rounds for decryption and encryption in which three rounds of encryption include byte substitution (sub bytes), shift row, mix columns, and add round key and the 4^{th} round uses three transformations only except for the mix columns. As for the decryption process, three rounds of decryption process include inverse shift row, inverse sub bytes, add round key and inverse mix columns, and the 4^{th} round uses three transformations only except for the inverse mix column. The algorithm of decryption and encryption is illustrated in Figure 1.

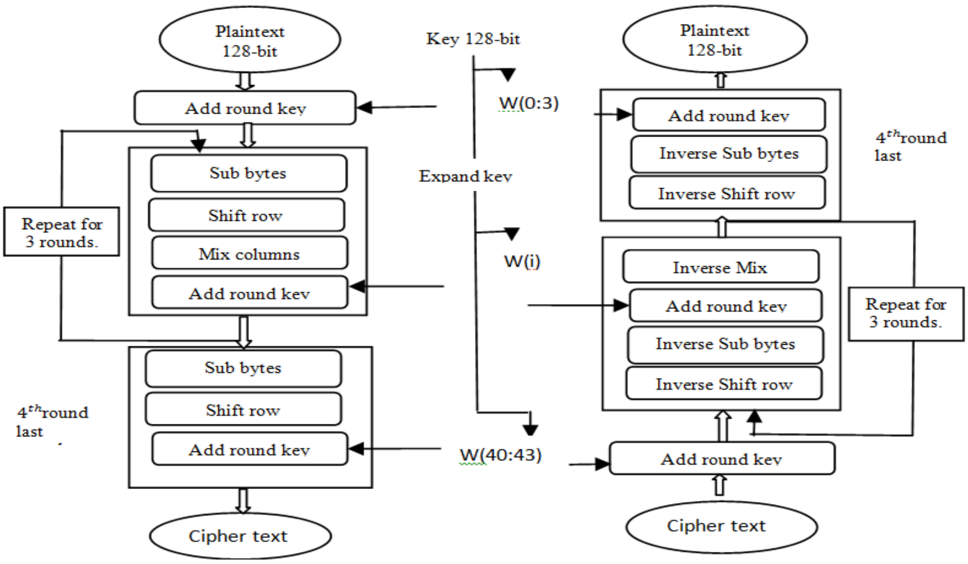


Figure 1. Algorithm for Encryption and decryption

We opted for four rounds of encryption and decryption based on experimentation, which revealed superior results to the standard AES with ten rounds. Implementing the AES involves utilizing both the standard AES S-box and the new S-box on the ESP32 card, chosen for its numerous advantages and capabilities. It consists of a 32-bit CPU, multiple general-purpose input/output (GPIO) ports, Bluetooth, Wi-Fi, and support for other protocols like serial peripheral interface (SPI), inter-integrated circuit protocol (I2C), and universal asynchronous receiver-transmitter (UART) are also included. It is established to support the IoT related projects [3]. A group of sensors was connected to represent the IoT part: the DH11 temperature and humidity sensor, ultrasound sensor, and LED lights, which were connected to the ESP32. Data is received by the ESP32 chip, which encrypts it before transmission to a dedicated internet page. Upon receiving the data on the page, it can be decrypted to unveil the actual values of the received information. This design is illustrated in Figure 2.

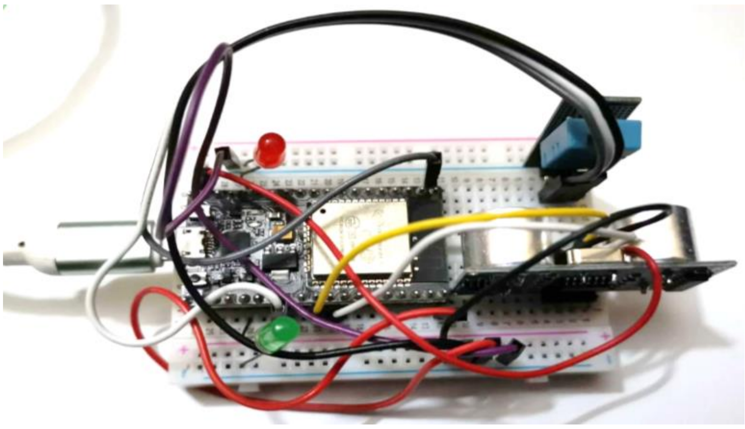


Figure 2. A proposed design using ESP32

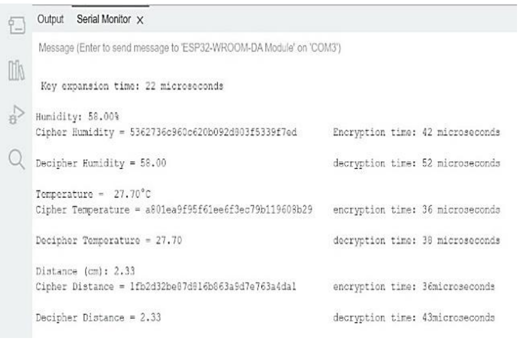
5. DISCUSSION AND RESULTS

The section includes evaluation of the avalanche effect and the time of decryption, encryption, and key expansion for both the standard AES and the new AES algorithms. The application is built on Arduino program. The cipher key utilized is “abcdefghijklmnop” during the experiments, and the plaintext was utilized as illustrated in Table 7 and Figures 3 and 4. The hardware required for the experiment includes an Intel Core i5-7200U CPU operating with 8 GB of RAM at 2.5 GHz and a 64-bit Windows 10 OS connected with the design in Figure 2. The study results demonstrate that the new AES algorithm works better than the standard AES128 in security and performance. The favorable outcomes hopefully will have a contribution in achieving high security for the IoT applications in the future.

5.1. Encryption and decryption time

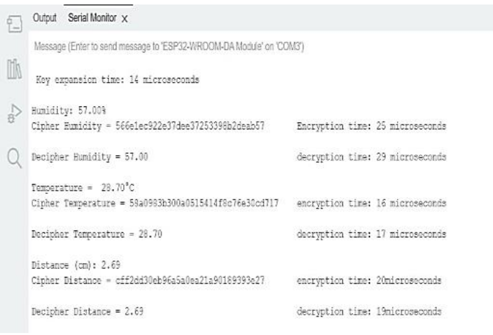
Encryption time means the duration needed to convert the plaintext to its corresponding ciphertext. In contrast, decryption time refers to the duration it takes to resume back the original plaintext from the ciphertext. While the time for key expansion is the duration used to build up a set of round keys from the original key. The number of rounds and the size of the key determine the number of round keys needed. Figures 3 and 4 represent the Arduino program’s serial interface, showing the sensor data, its value post encoding, its value after decoding to confirm the encryption algorithm is operating properly, the time of encryption and decryption process, and the key generation.

Every sample is encrypted and decrypted eight times to calculate the time performance of the plaintext cryptographic transformation. The average of eight experiments was taken as the encryption and decryption time. Based on these results, there is a 43.9% and 49.7% improvement in the encryption and decryption time, respectively, using the new AES. This enhancement in encryption and decryption time is due to the decrease in number of cipher and decipher rounds from ten to four rounds and the replacement of the standard S-box with the new S-box in the new AES algorithm. There is also a 36.36% improvement in the key expansion time using the New AES.



```
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')

 Key expansion time: 22 microseconds

Humidity: 58.00%
Cipher Humidity = 5362736c960c620b092d803f5339f7ed      Encryption time: 42 microseconds

Decipher Humidity = 58.00                               decryption time: 52 microseconds

Temperature =  27.70°C
Cipher Temperature = a801ea9f95f61ee6f3ec79b119608b29   encryption time: 36 microseconds

Decipher Temperature = 27.70                            decryption time: 38 microseconds

Distance (cm): 2.33
Cipher Distance = 1fb2d32be87d816b863a9d7e763a4da1      encryption time: 36microseconds

Decipher Distance = 2.33                                decryption time: 43microseconds
```

Figure 3. Output on Arduino serial for the standard S-box with ten rounds



```
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')

 Key expansion time: 14 microseconds

Humidity: 57.00%
Cipher Humidity = 566e1ec922e37dee37253398b2deab57      Encryption time: 25 microseconds

Decipher Humidity = 57.00                               decryption time: 29 microseconds

Temperature =  28.70°C
Cipher Temperature = 58a0983b300a0515414f8c76e30cd717   encryption time: 16 microseconds

Decipher Temperature = 28.70                            decryption time: 17 microseconds

Distance (cm): 2.69
Cipher Distance = cff2dd30eb96a5a0ea21a90189393e27      encryption time: 20microseconds

Decipher Distance = 2.69                                decryption time: 19microseconds
```

Figure 4. Output on Arduino serial for the new S-box with four rounds

5.2. Strict avalanche effect

Table 7 shows the avalanche effect of the standard AES and the new AES. One bit from the plaintext was changed in each test. The outcome demonstrates that the new AES achieved larger avalanche effect compared to the standard AES. Increased avalanche effect strengthens the security of the algorithms and enhances protection against cryptographic attacks and illegal access.

Table 7. SAC of new AES and standard AES with single bit change in plain text using the same key

Bit change pos.in plaintext	Plaintext	Standard AES ciphertext	NEW AES ciphertext	Standard AES-SAC	New AES -SAC
0	3243F6A8885A308D 313198A2E0370734	3925841D02DC09FB DC118597196A0B32	3C80B44313B44B57 DE5535607F534BAB	-	-
2	7243F6A8885A308D 313198A2E0370734	9E21ADE1135D57A7 F94567E065266306	AAA3C93E47312CB0 3614FFA811CDADD6	45.313	46.093
4	2243F6A8885A308D 313198A2E0370734	8D8BE5034154A5F3 4C71DC06274A73C0	C3974D38FAB7215F 5BAE6F4C54202926	40.625	53.906
8	3643F6A8885A308D 313198A2E0370734	E1D67C090EFC883A C94E311FD129D64C	D446190D47821778 2AFE690E692D6966	45.313	52.343
16	3242F6A8885A308D 313198A2E0370734	7747752A55AC3CA4 9F109AFEBFCD6833	8BC72AA4154E91F9 A742617A5F73AD29	48.437	52.343
32	3243F6A9885A308D 313198A2E0370734	2DBBDAB13E433C9 99C02259F3B0E2361	D35585934D4CAAF5 F9B25305DE2E3970	39.843	53.906
64	3243F6A8885A308C 313198A2E0370734	A59AA956AAF298C2 9C672652C880EDD2	F5A1E80DC91DEB12 9AB7ADD797A0F390	50	48.437
128	3243F6A8885A308D 313198A2E0370735	30A25D6A5C95DDE 2390758B150FF7038	3FE1174B94855AF0 8B3A274B3EB53725	45.31	49.219

Figure 5 shows an increase in the results of the avalanche effect of the new AES on the sample of plaintexts used compared with the standard AES. The new AES achieved 50.89%, greater than the ideal value of 50%, however the standard AES achieved 44.97%, according to the sample data set. Therefore, the new AES outperforms the standard AES by 11.63%.

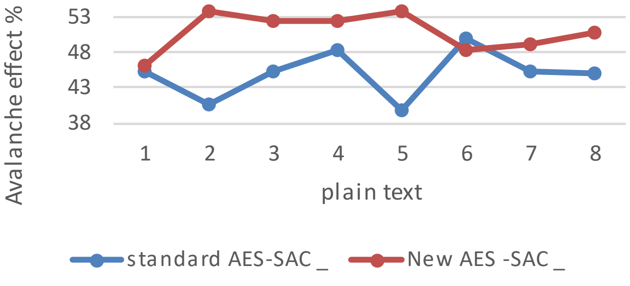


Figure 5. Avalanche effect for new AES and standard AES

6. CONCLUSION

This paper introduces an enhanced version of AES algorithm which utilizes a new S-box with fewer rounds. This modification aims to enhance plaintext encryption and decryption security against potential hackers. The modified algorithm demonstrates stronger cryptographic strength than the standard AES, as evidenced by the avalanche effect results. So, by reducing the rounds number and using new S-box instead of standard S-box, the time required for key expansion, decryption, and encryption has decreased, and the memory size used has been reduced. Consequently, the new enhanced algorithm with four rounds of encryption with the new S-box achieved better results than the standard S-box with ten rounds and is more resistant to algebraic attacks. Additionally, we have designed an IoT-based smart environment to test whether the proposed method is secure from numerous vulnerabilities. We hope our implementation can be effective in many valuable fields, for example, inside neonatal intensive care units or inside nuclear reactors. For future work, the efficacy of the new AES algorithm in encrypting various types of data, including audio and video files, will be evaluated, and compared with the standard AES algorithm.

REFERENCES

[1] A. Goulart, A. Chennamaneni, D. Torre, B. Hur, and F. Y. Al-Aboosi, "On wide-area iot networks, lightweight security and their applications—a practical review," *Electronics*, vol. 11, no. 11, p. 1762, Jun. 2022, doi: 10.3390/electronics11111762.

[2] P. Pukkasenung, "Internet of things (IoT): a basic concept and analysis security issues," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 18, no. 11, pp. 1–10, 2020, doi: 10.5281/zenodo.4361756.

[3] M. Al-Mashhadani and M. Shujaa, "IoT security using AES encryption technology based ESP32 platform," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 214–223, 2022, doi: 10.34028/iajit/19/2/8.

[4] R. Florin and R. Ionut, "FPGA based architecture for securing IoT with blockchain," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Oct. 2019, pp. 1–8, doi: 10.1109/SPED.2019.8906595.

[5] V. V. D. Reshma M Nadaf, "Hardware implementation of modified AES with key dependent dynamic S-box," *IEEE Icaret*, pp. 576–580, 2012.

[6] H. A. M. A. Basha, A. S. S. Mohra, T. O. M. Diab, and W. I. El-Sobky, "Efficient image encryption based on new substitution box using DNA coding and bent function," *IEEE Access*, vol. 10, pp. 66409–66429, 2022, doi: 10.1109/ACCESS.2022.3183990.

[7] F. A. Kadhim, G. H. A. Majeed, and R. S. Ali, "Proposal new S-box depending on DNA computing and mathematical operations," in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, May 2016, pp. 1–6, doi: 10.1109/AIC-MITCSA.2016.7759926.

[8] N. Siddiqui, A. Naseer, and M. Ehatisham-ul-Haq, "A novel scheme of substitution-box design based on modified pascal's triangle and elliptic curve," *Wireless Personal Communications*, vol. 116, no. 4, 2021, doi: 10.1007/s11277-020-07832-y.

[9] U. Hayat, N. A. Azam, H. R. Gallegos-Ruiz, S. Naz, and L. Batool, "A truly dynamic substitution box generator for block ciphers based on elliptic curves over finite rings," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8887–8899, Sep. 2021, doi: 10.1007/s13369-021-05666-9.

[10] L. C. Nizam Chew and E. S. Ismail, "S-box construction based on linear fractional transformation and permutation function," *Symmetry*, vol. 12, no. 5, p. 826, May 2020, doi: 10.3390/sym12050826.

[11] A. H. Zahid *et al.*, "Efficient dynamic S-box generation using linear trigonometric transformation for security applications," *IEEE Access*, vol. 9, pp. 98460–98475, 2021, doi: 10.1109/ACCESS.2021.3095618.

[12] Z. Gan, X. Chai, J. Zhang, Y. Zhang, and Y. Chen, "An effective image compression–encryption scheme based on compressive sensing (CS) and game of life (GOL)," *Neural Computing and Applications*, vol. 32, no. 17, 2020, doi: 10.1007/s00521-020-04808-8.

[13] Y.-G. Yang, B.-W. Guan, J. Li, D. Li, Y.-H. Zhou, and W.-M. Shi, "Image compression-encryption scheme based on fractional order hyper-chaotic systems combined with 2D compressed sensing and DNA encoding," *Optics & Laser Technology*, vol. 119, p. 105661, Nov. 2019, doi: 10.1016/j.optlastec.2019.105661.

[14] E. W. Afify *et al.*, "Performance analysis of advanced encryption standard (AES) S-boxes," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 1, pp. 2214–2218, 2020, doi: 10.35940/ijrte.f9712.059120.

[15] K. Z. Zamli, F. Din, and H. S. Alhadawi, "Correction to: exploring a Q-learning-based chaotic naked mole rat algorithm for S-box construction and optimization," *Neural Computing and Applications*, vol. 35, no. 19, 2023, doi: 10.1007/s00521-023-08416-0.

[16] E. Corona-Bermúdez, J. C. Chimal-Eguía, U. Corona-Bermúdez, and M. E. Rivero-Ángeles, "Chaos meets cryptography: developing an S-box design with the rössler attractor," *Mathematics*, vol. 11, no. 22, 2023, doi: 10.3390/math11224575.

[17] A. Razaq, G. Alhamzi, S. Abbas, M. Ahmad, and A. Razzaque, "Secure communication through reliable S-box design: a proposed approach using coset graphs and matrix operations," *Heliyon*, vol. 9, no. 5, 2023, doi: 10.1016/j.heliyon.2023.e15902.

[18] S. Zhou, Y. Qiu, X. Wang, and Y. Zhang, "Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box," *Nonlinear Dynamics*, vol. 111, no. 10, pp. 9571–9589, 2023, doi: 10.1007/s11071-023-08312-1.

[19] P. Tian and R. Su, "A novel virtual optical image encryption scheme created by combining chaotic S-box with double random phase encoding," *Sensors*, vol. 22, no. 14, p. 5325, Jul. 2022, doi: 10.3390/s22145325.

[20] A. Alhudhaif, M. Ahmad, A. Alkhayyat, N. Tsafack, A. K. Farhan, and R. Ahmed, "Block cipher nonlinear confusion components based on new 5-D hyperchaotic system," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3090163.

[21] A. H. Zahid *et al.*, "Dynamic S-box design using a novel square polynomial transformation and permutation," *IEEE Access*, vol. 9, pp. 82390–82401, 2021, doi: 10.1109/ACCESS.2021.3086717.

[22] M. Ahmad and E. Al-Solami, "Evolving dynamic s-boxes using fractional-order hopfield neural network based scheme," *Entropy*, vol. 22, no. 7, p. 717, Jun. 2020, doi: 10.3390/e22070717.

[23] S. Ibrahim *et al.*, "Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps," *IEEE Access*, vol. 8, pp. 160433–160449, 2020, doi: 10.1109/ACCESS.2020.3020746.

[24] A. Razaq *et al.*, "A Novel method for generation of strong substitution-boxes based on coset graphs and symmetric groups," *IEEE Access*, vol. 8, pp. 75473–75490, 2020, doi: 10.1109/ACCESS.2020.2989676.

[25] D. Lambić, "A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design," *Nonlinear Dynamics*, vol. 100, no. 1, pp. 699–711, Mar. 2020, doi: 10.1007/s11071-020-05503-y.

[26] M. Ahmad, E. Al-Solami, A. M. Alghamdi, and M. A. Yousaf, “Bijective S-boxes method using improved chaotic map-based heuristic search and algebraic group structures,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3001868.

[27] F. Özkaynak, “On the effect of chaotic system in performance characteristics of chaos based S-box designs,” *Physica A: Statistical Mechanics and its Applications*, vol. 550, p. 124072, Jul. 2020, doi: 10.1016/j.physa.2019.124072.

[28] A. M. Abdullah, “Advanced encryption standard (AES) algorithm to encrypt and decrypt data,” *Cryptography and Network Security*, no. June, 2017, [Online]. Available: https://www.researchgate.net/publication/317615794.

[29] A. R. Chowdhury, J. Mahmud, A. R. M. Kamal, and M. A. Hamid, “MAES: modified advanced encryption standard for resource constraint environments,” in *2018 IEEE Sensors Applications Symposium (SAS)*, 2018, pp. 1–6, doi: 10.1109/SAS.2018.8336747.

[30] S. Hassan and M. A. Zaid, “Modification advanced encryption standard for design lightweight algorithms,” *Journal of Kufa for Mathematics and Computer*, vol. 6, no. 1, May 2019, doi: 10.31642/JoKMC/2018/060104.

[31] S. S. Rekha and P. Saravanan, “Low-cost AES-128 implementation for edge devices in IoT applications,” *Journal of Circuits, Systems and Computers*, vol. 28, no. 04, p. 1950062, Apr. 2019, doi: 10.1142/S0218126619500622.

[32] M. S. Fadhil, A. K. Farhan, M. N. Fadhil, and N. M. G. Al-Saidi, “A new lightweight AES using a combination of chaotic systems,” in *2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA)*, Jul. 2020, pp. 82–88, doi: 10.1109/IT-ELA50150.2020.9253099.

[33] B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, and A. Alzamil, “Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic S-box,” *Symmetry*, vol. 13, no. 1, p. 129, Jan. 2021, doi: 10.3390/sym13010129.

[34] Z. Rahman, X. Yi, M. Billah, M. Sumi, and A. Anwar, “Enhancing AES using chaos and logistic map-based key generation technique for securing IoT-based smart home,” *Electronics*, vol. 11, no. 7, p. 1083, Mar. 2022, doi: 10.3390/electronics11071083.

[35] D. N. Hammod, “Modified lightweight AES based on replacement table and chaotic system,” in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022, pp. 1–5, doi: 10.1109/HORA55278.2022.9799804.

[36] Institute of Electrical and Electronics Engineers, Guo li jiao tong da xue (Taiwan). “Taiwan Information Security Center., and IEEE Reliability Society.,” *2017 IEEE Conference on Dependable and Secure Computing: Taipei, Taiwan, August 7-10, 2017*.

[37] J. H. Cheon and D. H. Lee, “Resistance of S-boxes against algebraic attacks,” in *Fast Software Encryption: 11th International Workshop, FSE 2004*, 2004, vol. 3017, pp. 83–93, doi: 10.1007/978-3-540-25937-4_6.

[38] A. F. Webster and S. E. Tavares, “On the design of S-boxes,” in *Conference on the theory and application of cryptographic techniques*, 1986, vol. 218 LNCS, pp. 523–534, doi: 10.1007/3-540-39799-X_41.

[39] E. W. Afify, W. I. El-Sobky, A. Twakol, and R. A. Alez, “Algebraic construction of powerful substitution box,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 405–409, 2020, doi: 10.35940/ijrte.d8279.038620.

[40] L. Jinomeiq, W. Baoduui, and W. Xinmei, “One AES S-box to increase complexity and its cryptanalysis,” *Journal of Systems Engineering and Electronics*, vol. 18, no. 2, pp. 427–433, 2007, doi: 10.1016/s1004-4132(07)60108-x.

[41] J. Cui, L. Huang, H. Zhong, C. Chang, and W. Yang, “An improved AES S-box and its performance analysis,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 5 A, pp. 2291–2302, 2011.

[42] A. Nitaj, W. Susilo, and J. Tonien, “A new improved AES S-box with enhanced properties,” in *Information Security and Privacy: 25th Australasian Conference, ACISP 2020*, 2020, vol. 12248 LNCS, pp. 125–141, doi: 10.1007/978-3-030-55304-3_7.

BIOGRAPHIES OF AUTHORS

Shaimaa S. Saleh received the B.Sc. degree in electronics and communications from Benha Faculty of Engineering in 2010 and the M.S. degree in 2016 from Benha University, Egypt. She is currently a teaching assistant at Benha Faculty of Engineering, Benha University, Egypt and pursuing the Ph.D. degree. Her current research interest in IoT security. She can be contacted at email: shimaa.said@bhit.bu.edu.eg.

Amr A. Al-Awamry is an Associate Professor at the Faculty of Engineering in Benha University. He received his Ph.D. degree from St. Petersburg Electrotechnical University, Russia in 2012, his M.Sc. from Communication department in Benha Faculty of Engineering Egypt in 2005 and the B.Sc. degree in Communication Engineering–Benha Faculty of Engineering, Egypt, in 1999. He can be contacted at email: amr.awamry@bhit.bu.edu.eg.

Ahmed Taha received his M.Sc. degree and his Ph.D. degree in Computer Science, at Ain Shams University, Egypt, in February 2009 and July 2015. He currently works as an Associate Professor at the Computer Science Department, Benha University, Egypt. He is the founder and coordinator of “Information Security and Digital Forensics” program, Faculty of Computers and Artificial Intelligence, Benha University. His research interest’s concern: computer vision and image processing (human behavior analysis-video surveillance systems), digital forensics (image forgery detection–document forgery detection), security (encryption–steganography–cloud computing), content-based retrieval (arabic text retrieval-video scenes classification-video scenes retrieval–trademark image retrieval-closed-caption technology). He can be contacted at email: ahmed.taha@fci.bu.edu.eg.