DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

Designing a New Lightweight AES Algorithm to Improve the Security of the IoT Environment

Sameeh Abdulghafour Jassim[1], Alaa K. Farhan[2]
[1,2]*Department of Computer Sciences, University of Technology, Baghdad, Iraq*
[1]*cs.19.22@grad.uotechnology.edu.iq*, [2]*Alaa.K.Farhan@uotechnology.edu.iq*

Abstract— *Recently, the Internet of Things (IoT) is begin used in many fields such as smart homes, healthcare systems, industrial applications, etc. Therefore, the use of the IoT led to a growth in the number of dangers especially in the areas of privacy and security for applications running on low-resource computers. Consequently, the demand for lightweight encryption methods is growing. To safeguard sensing data, this study introduces a Lightweight Advanced Encryption Standard (LAES) depending on dynamic ShiftRows, initial permutation instead of MixColumns, and a dynamic number of rounds. It was created with the goal of reducing encryption/decryption time. The proposed approach was assessed by using various measurements such as lengths of the key used was 2^{128} and it is quite enough for security, key sensitivity values were 100%, Also, this study compared the encryption/decryption time, NIST statistical test, and security strength of the proposed architecture to those of XTEA, SIMON, Skinny, SPECK, and PRESENT. The encryption/decryption time of the proposed approach was had the shortest period (0.0169 S) while the SPECK algorithm was had the longest period (4.1249 S) among the comparative algorithms. Whereas, NIST statistical test values of the proposed approach were passed successfully and had higher values than the comparative algorithms. Moreover, the proposed approach utilized 1280, 1024, and 768 GE with 6, 8, or 10 rounds respectively. The average number of GE was approximately 1000 GE. These numbers of GE are considered highly efficient with the IoT environment.*

Index Terms— *IoT, AES, Lightweight systems, Chaotic systems, Cryptography*.

I. INTRODUCTION

IoT is becoming increasingly popular and widespread because of its vast variety of applications in many fields. They gather data from the actual world and send it through the internet. When it comes to deploying IoT in the real world, there are numerous challenges to overcome, ranging from small sensors to servers. Because most devices of IoT could be accessed physically in the real world and most of them are limited in resources (such as processing power, energy, bandwidth, memory), security is regarded as the most significant problem in IoT deployments [1]. Several industry professionals and scholars have been defining the IoT in numerous ways, based on their implementation areas and applications. But, in simpler terms, the IoT is a network of linked devices, each one of them has a unique identifier, that can gather and share data via the Internet with or without human intervention [2]. The devices of the IoT may be split into two groups: devices have a lot of resources, such as smartphones, servers, tablets, and personal computers, etc. and devices have limited resources, such as actuators, sensor nodes, or industrial sensors, RFID tags, and so on [3] [4]. Moreover, cryptography may be one of the most effective methods for ensuring the

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

integrity, confidentiality, authorization, and authentication of data passing via IoT devices. It might also be a way to protect data that is stored or sent via a network. Therefore, this paper focuses on designing a lightweight AES that meets the requirements of constraints IoT devices as well as providing a high level of security. The following are the major contributions of this work:

- Proposed a new method of lightweight AES (LAES) that is lightweight, fast, and secure.
- The dynamic Shift Rows are utilized instead of the traditional Shift Rows.
- To reduce the processing time and eliminate the MixColumns' complicated multiplication procedures. This study utilized Initial Permutation (IP) instead of MixColumns.
- The two S-boxes of the IPs are built using two chaotic systems (quadratic and piecewise).
- The secret keys are constructed using a chebyshev map.
- Utilizing a dynamic number of rounds.
- Tested all results of the proposed approach with various measurements.

The remainder of the paper is structured as follows. Section two provides related work of the lightweight algorithms. Section three describes the system background. The proposed approach is presented in section four. Section five assesses the proposed architecture's performance and compares it to existing approaches. Finally, the most important concludes are discussed in section six.

II. RELATED WORK

Security is one of the most important challenges in the IoT. Many researchers have worked to solve security concerns, and the following paragraphs highlight their contributions.

In [5] the Extended Tiny Encryption Algorithm (XTEA) was proposed. XTEA is a block cipher that encrypts/decrypts data in 64-bit blocks using a cryptographic key of 128 bits. It utilized a Feistel Network (FN) and is suited for low resource environments.

In [6] SIMON was designed by NSA. SIMON is recognized for its minimal hardware footprint. It provides multiple key sizes (64, 72, 96, 128, 144, 192, 256) bits across the block of (32, 48, 64, 96, 128, 144, 192, 256, 192, 256) bits over (32, 36, 42, 44, 52, 54, 68, 69, 72) rounds. Execution of the most compact version needs the use of 763 GE.

In addition, the two versions of Skinny (SKINNY-64 and SKINNY-128) are presented also in [6]. SKINNY-64 does (32, 36, 40) rounds with a (64, 128, 192) bit key and 64-bit block, whereas SKINNY-128 performs (40, 48, 56) rounds with a (128, 256, 384) bit key and 128-bit block.

In [7] SPECK is software-oriented encryption that is a sibling of SIMON and was created by NSA. It can do (22, 23, 26, 27, 28, 29, 32, 33, and 34) iterations with the same block and key sizes as SIMON. The smallest hardware implementation reported utilizes an 884 GE, using a 48-bit block with a 96-bit key, while the implementation of the most efficient software takes 186 bytes of ROM and 599 cycles for a 64-bit block with a 128-bit key, requiring 599 cycles and 186 bytes of ROM.

In [8] PRESENT is built on an SPN and employs 64-bit blocks on variants of two key: (128-bit) and (80-bit) keys, with GE demands of 1886 and 1570, respectively.

All related works paragraphs of the different lightweight block ciphers algorithms (XTEA, SIMON, Skinny, SPECK, and PRESENT) mentioned previously were most

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

popular used. Therefore, the results of these works were used to compare with the proposed technique results.

III. SYSTEM BACKGROUND

The background of this work will cover the AES algorithm, Chaos theory, and Lightweight block cipher.

A. AES Algorithm

The AES technique is widely used for data encryption. Its work depends on the substitution/permutation/network (SPN) [9]. Moreover, AES is an encryption algorithm that uses symmetrical blocks [10]. All of the algorithm's operations are 8-bit or higher. The cipher block uses plaintext of sizes (128, 192, 256) bits for (10, 12, 14) rounds respectively [11]. Encryption and decryption keys are represented as a square matrix of bytes. The method of AES encryption is depicted in *Fig. 1* [12].

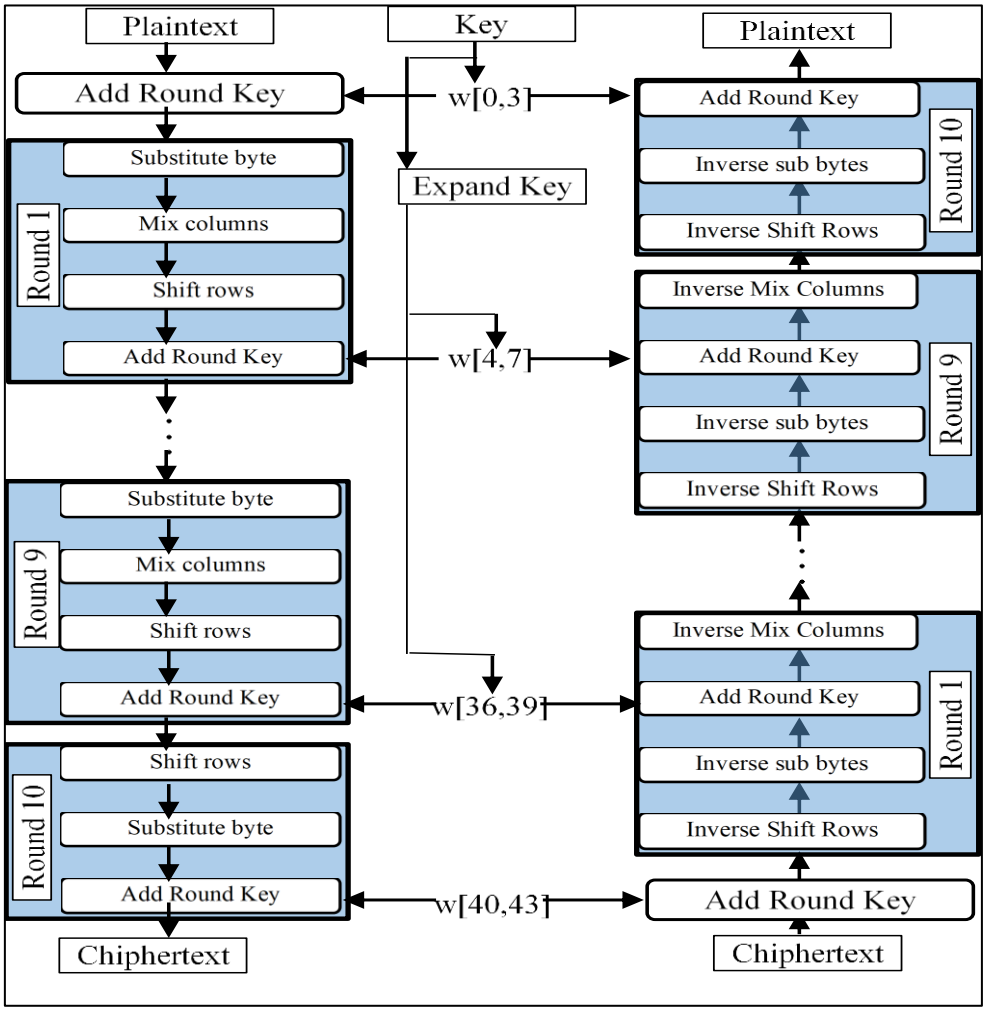


FIG. 1. AES ALGORITHM ARCHITECTURE.

B. The Theory of Chaos

Chaos theory can be defined as a branch of mathematics that deals with deterministic and nonlinear behavior [13]. The output signals of chaotic systems are random and unpredictable [14]. It is more sensitive to its initial conditions (control parameters and

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

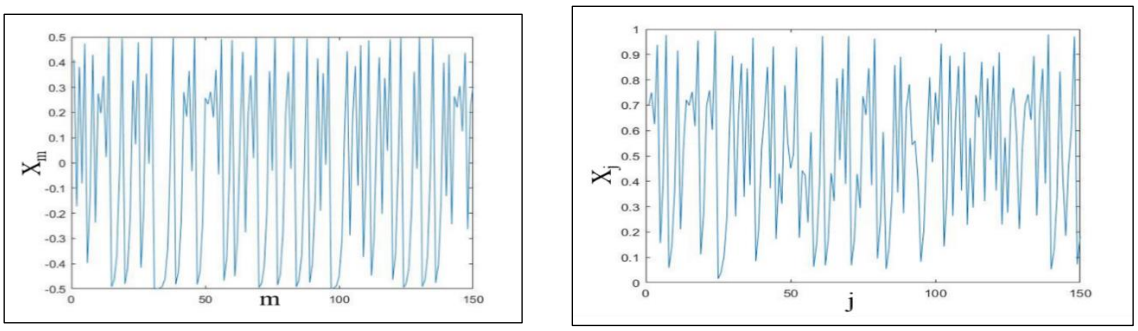
initial values); as a result, a slight modification in the initial conditions causes a significant modification in the productions of the chaotic values [15]. The characteristics of the chaotic systems are related to the confusion and diffusion property of a good cipher in cryptography, which has led numerous researchers to employ chaotic theory to improve various cryptographic systems' security [16] [17]. This paper used two chaotic equations: quadratic map and piecewise map. The quadratic map is a well-known chaotic map with complicated dynamic behavior is the quadratic map. In cryptography applications, this map has been frequently utilized. as the following Eq.1 [18]:

$$x_{m+1} = k - (x_m)^2 \quad (1)$$

where, k with a range of [0,2] is a control parameter, m is the number of iterations, and X_m [0, 1] is the generated chaotic sequence. As shown in *Fig. 2.* (a). Whereas, piecewise map that used in this paper (as shown in *Fig.* 2. (b)) is illustrated in Eq. (2) [19]:

$$X_{j+1} = \begin{cases} \frac{X_j}{R} & 0 \le X_j \le R \\ \frac{X_j - R}{0.5 - R} & R \le X_j < 0.5 \\ \frac{1 - R - X_j}{0.5 - R} & 0.5 \le X_j < 1 - R \\ \frac{1 - X_j}{R} & 1 - R \le X_j < 1 \end{cases} \quad (2)$$

where, $R \in (0,0.5) and\ R \neq 0$.



(A) QUADRATIC (B) PIECEWISE

FIG. 2. TWO CATEGORIES OF CHAOTIC MAPS.

C. Lightweight Block Cipher

There are many definitions available in the literature for IoT. According to [20], IoT is made up of two words: "Internet" and "Things" resulting in two distinct visions. The first is focused on the 'Internet' or network component, while the second is focused on the 'things' component. sensors, actuators, and RFID tags are referred to as "things". Accordingly, these things must interoperate within many privacy and security issues, for example, secure communication, confidentiality, etc. These things manage critical and private information in various ways. As a result, they must provide enough protection to prevent multiple hackers/attacks [9]. A block cipher can be defined as a form of symmetric encryption that processes a whole block at one time [21]. Feistel block ciphers and SPN are two forms of lightweight block ciphers. SPN is simpler, but it lacks a schedule of keys, so it will be susceptible to attacks. While only half of the state is utilized by the Feistel structure's round function. Thus, the result utilized the same code for the encryption/decryption operations. Moreover, It utilized less memory and it could be running on hardware with low power [22]. The number of rounds, key size, structure type, and block size are the most important factors to consider while evaluating a lightweight block cipher. There are different

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

lightweight block ciphers algorithms, and the most popular used are XTEA, SIMON, Skinny, SPECK, and PRESENT. In addition, the lightweight stream cipher was illustrated in [23].

IV. THE PROPOSED METHOD OF LIGHTWEIGHT AES

The fundamental objective of this proposal is to provide an AES encryption method that is lightweight, fast, and secure. It may be utilized to secure data from IoT sensors. To accomplish the lightweight algorithm, there are several improvements are made to the structure of the AES algorithm. This paper proposed lightweight AES depending on exploiting a variety of chaotic theories to produce an initial permutation instead of MixColumns and provide an encryption system that is highly safe, as illustrated in *Fig. 3*. The suggested lightweight AES performs the same functions as the traditional AES with some improvements:

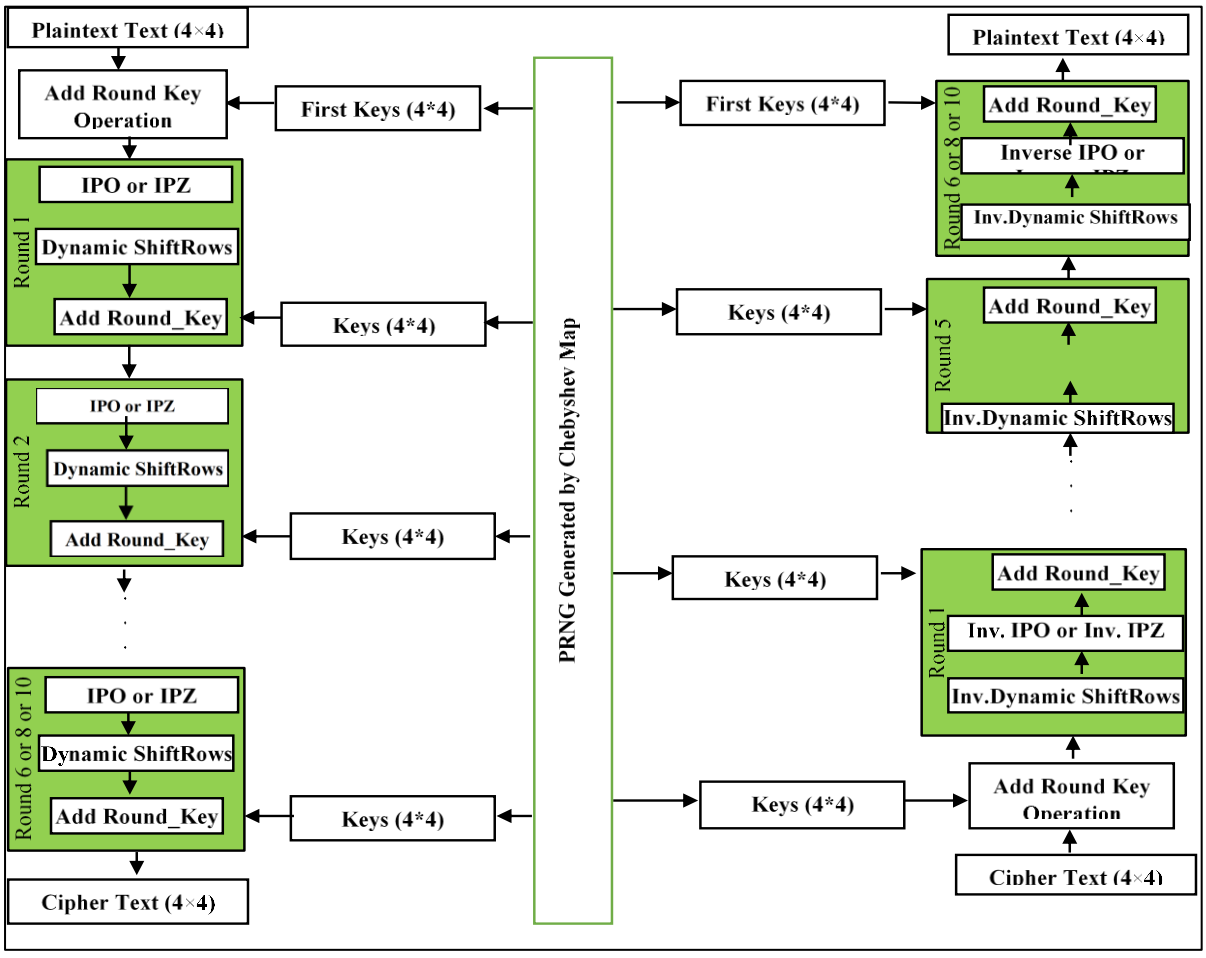


FIG. 3. THE PROPOSED LAES ALGORITHM ARCHITECTURE.

Hence, the dynamic Shift Rows are utilized rather than the traditional Shift Rows. Also, instead of MixColumns, the Initial Permutation (IP) is utilized. This improvement reduces the processing time and eliminates the MixColumns' complicated multiplication procedures. The two S-boxes of the IPs are built using two chaotic systems (quadratic and piecewise). The secret keys are constructed using a chebyshev map.

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

A. Key Generation of System

Algorithm 1 includes the construction of secret keys values (PRNG). Chebyshev map is used to produce the secret keys utilized in the encryption operation. The chebyshev map is a type of chaotic one-dimensional map that has the following definition:

$$x_j + 1 = abs(\cos(k \times cos^{-1} x_j)) \quad (3)$$

where the control parameter k ∈ J; j is a non-negative integer; k > 1.

Algorithm 1: processing of chebyshev map

1: **Input**: master key;
2: **Output**: distinct random numbers;
3: Begin
4: The control parameter is c ∈ [0,1]
5: For j = 0 to 255
6: $x_j + 1 = abs(\cos(k \times cos^{-1} x_j))$
7: The step 6 results are converted to integer numbers $(int(x_j + 1 \times 255))$;
8: End For
9: End

Hence, duplicate numbers resulting from Algorithm 1 are deleted and replaced randomly with new values that do not exist in the output.

B. The Proposed Initial Permutation

The IP operation is used instead of MixColumns to give diffusion. This improvement was used to eliminate the MixColumns' complicated multiplication procedures. Table I. (a) and Table II. (a) of IP is created utilizing the two chaotic maps quadratic and piecewise respectively. Two IP tables are created, one used when the master key is one (IPO) and the other users when the master key is zero (IPZ). Both of them with a size of (4×4) to match the state (4×4). In addition, the inverse of IPO and the inverse of IPZ are shown in Table I. (b) and Table II. (b) respectively. While, the constructing of IPs is represented by Algorithm 2, and the inverse constructing of IPs is represented by Algorithm 3.

Algorithm 2: construction of IPs

1: **Input**: master key (M) used as initial values for both Eq.1 and Eq.2;
2: **Output**: IPO (4×4) is used when the master key is one, and IPZ (4×4) is used when the master key is zero;
3: Begin
4: Use Eq.1 and Eq.2 to generate random numbers.
5: Multiplying the results of step 4 and round it to integer numbers.
6: Eliminate the duplicated values from step 5.
7: Add randomly new values [0,15] that do not exist in the output.
8: End

TABLE I. IPO AND IPZ

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

(a) IPO

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

(b) IPZ

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

TABLE II. INVERSE IPO AND INVERSE IPZ

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

(a) inverse IPO

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

(b) inverse IPZ

Algorithm 3: construction of inverse IPs

1: **Input**: IPO (4×4) is used when the master key is one, and IPZ (4×4) is used when the master key is zero;
2: **Output**: Inv. IPO (4×4) and inv. IPZ (4×4);
3: Begin
4: Loop through *IP*.
5: In inverse IP, the IP content becomes the numbers index, and the numbers index in IP becomes the content.
6: Steps 4 to 5 are implemented on IPO and IPZ to build both inverse IPO and inverse IPZ respectively.
7: End

C. Dynamic Shift Rows

In this work, the dynamic Shift Rows are utilized instead of the traditional Shift Rows. As illustrated in Algorithm 2. the number of rotations in the state array depends on the master key. The shift row procedures in both encryption and decryption operations are the same.

Algorithm 4: Dynamic Shift Rows

1: **Input**: master key (M);
2: **Output**: new array;
3: Begin
4: Convert M to a binary number (BM);
5: For each row:
6: For j = 0 to 8 step:2
7: if BM[j]== ‘00’ then
8: State_array= No Rotate
9: Elseif BM[j]== ‘01’ then
10: State_array= Rotate the array once
11: Elseif BM[j]== ‘10’ then
12: State_array= Rotate the array twice
13: Elseif BM[j]== ‘11’ then
14: State_array= Rotate the array three times
15: End For
16: End for
17: End

D. Dynamic Number of Rounds

In the traditional AES algorithm, the cipher block uses plaintext of sizes (128, 192, 256) bit for (10, 12, 14) rounds respectively. Consequently, those numbers of rounds are heavy and unsuitable with the constraint devices. Therefore, this paper suggested a new technique depending on a dynamic number of rounds with the plaintext of sizes 128-bit. As

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

illustrated in Algorithm 5. Hence, this study reduced the number of rounds while the complexity of the AES algorithm was increased. Because the number of rounds will be changed in each new block depending on the master key.

Algorithm 5: Dynamic Number of Rounds

1: **Input**: master key M;
2: **Output**: dynamic number of rounds;
3: Begin
4: Convert M to a binary number (BM);
5: Read every two bits of BM separately until completed the required numbers of rounds.
6: Case BM== '00' then "Ignore this case and read another two digits";
7: Case MB=='01' then the numbers of the round are six;
8: Case MB=='10' then the numbers of the round are eight;
9: Case MB=='11' then the numbers of the round are ten;
10: Repeat step 5 to step 9 for each new block until complete encrypted the plaintext.
11: End.

E. The Encryption\ Decryption processes

As shown in *Fig. 3*, each round of the suggested lightweight AES includes the operations IP, Dynamic ShiftRows, and AddRoundKey. These operations are represented by Algorithm 6.

Algorithm 6: the encryption

1: **Input**: master key, plaintext;
2: **Output**: the ciphertext;
3: Begin
4: partition plaintext into 16-byte blocks and transform each block to a state array with a size of (4×4);
5: add round (the key with a state of (4×4) generated by algorithm 1).
6: used algorithm 5 to determine the number of rounds.
7: used algorithm 2 to apply either IPO (4×4) or IPZ (4×4) depending on the master key.
8: used algorithm 4 to apply dynamic shift rows depending on the master key.
9: save the encrypted data and repeat steps 5 to 8 with new keys on the next portion of the plaintext block.
10: End.

While in the decryption process all steps in Algorithm 6 are applied in a reverse way except in step 7 the Algorithm 3 is applied instead of Algorithm 2. As shown in *Fig. 3*.

V. RESULTS AND DISCUSSION

This part explained the sizes of block and lengths of key, key sensitivity, and comparative analysis, respectively. All tests were conducted on an HP laptop with an Intel(R) CoreTM i7-8565U processor operating at 3.79 GHz and 8 GB of RAM, running Windows 10 (64-bit OS) with Python 3.9.0 and Tk 8.6.9.

A. Sizes of Block and Lengths of Key

In cryptography, a block can be defined as data with a fixed sized piece that is to be processed by an algorithm at the same time. In this work, the size of blocks is 16 bytes (which is the same as 128-bit). Furthermore, to contain all of the plain texts, encrypted files

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

are partitioned into many blocks. Padding is used to fill the remaining space when the final block of plaintext is not filled. While the length of a password/key in a cryptographic system relates to how many bits it has [24]. In the proposed approach, the key length used was 2^{128} is quite enough for security. Since the efficient key space should be larger than 2^{100} [25]. Therefore, the proposed approach can avoid attacks of the brute-force.

B. The Sensitivity of the Key

The sensitivity of the key in the cryptosystem field must be as high as possible to avoid attacks of brute force. The influence of a slight change in the secret keys on the encryption/decryption operations is determined by key sensitivity. The cipher-text difference rate (CDR) is the rate at which the cipher text changes when the secret key is modified slightly. The CDR equation is as follows [26]:

$$CDR = \frac{diff(CH,CH1)+diff(CH,CH2)}{2\times S} \times 100\% \quad (4)$$

where S is the size of the cipher text. CH, CH1, and CH2 are plaintext that has been encrypted with the secret keys SK, SK+ΔSK, and SK - ΔSK, respectively. ΔK is the tiny difference in secret keys, and the number of values that differ between the encrypted plaintext CH and CH1 is diff(CH, CH1). The proposed approach was tested by using various data sizes (256, 512, 1024, 2048, and 4096 bytes) with different encryption keys. Consequently, the key sensitivity CDR was 100%. As a result, even little changes in the key generate significant changes in the cipher text.

C. Comparative Analysis

This part compares the encryption/decryption time, NIST statistical test, and security strength of the proposed architecture to those of XTEA, SIMON, Skinny, SPECK, and PRESENT.

1. Encryption/decryption time

The time it takes to convert plaintext to ciphertext is known as encryption time. Whereas decryption time is the time it takes to transform ciphertext to plain text. Encryption/decryption time is a measurement of how long it takes to complete the encryption/decryption process. An inefficient technique is indicated by a long encryption/decryption time. As illustrated in *Fig. 4*, existing techniques such as XTEA, SIMON, Skinny, SPECK, and PRESENT take longer encryption/decryption time than the suggested approach. Consequently, the proposed approach reduces the time it takes to the encryption/decryption process compared to other techniques.

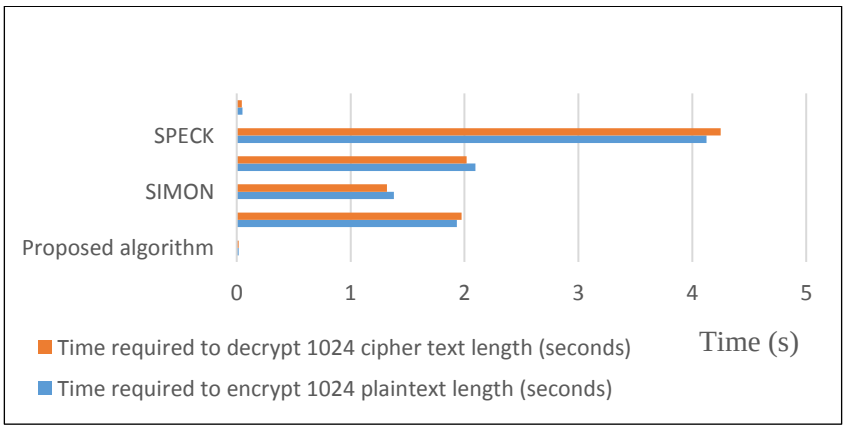


FIG. 4. COMPUTED ENCRYPTION/DECRYPTION TIME COMPARISON (SECONDS).

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

2. Statistical Analysis

To confirm that the number of sequences produced by the encryption resembles pseudorandom sequences with no statistical flaws, the NIST statistical test [27] was employed. The number sequences were created by encrypting plaintext (1024-byte sample) with a fixed key. The suggested method does certainly approximate a pseudorandom function, which is one of the desirable features of a block cipher, as shown in Table II. Furthermore, the comparative algorithms had lower values than the proposed approach except the present algorithm was has results values close to the proposed approach. [note that: all results in Table III were tested by the authors].

TABLE III. RESULT OF NIST TEST

Name of test		PRESENT	SPECK	Skinny	SIMON	XTEA	The Proposed algorithm
Test of overlapping template of all one's		1.000000	FAILURE	1.000000	FAILURE	1.000000	1.000000
Test of nonperiodic templates		1.000000	FAILURE	1.000000	FAILURE	1.000000	1.000000
Test of Frequency within a block		1.000000	0.898710	0.595883	0.908730	0.051830	0.828880
The longest run of ones in a block test		1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Test of frequency		0.802587	0.802587	0.595883	0.453255	0.051830	0.488844
Test of discrete fourier transform (spectral)		0.194366	0.745603	0.491297	0.256145	0.301898	0.908677
Test of approximate entropy		1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Test of Runs		0.457534	0.796357	0.704505	0.492893	0.598506	0.525199
Test of cumulative sums	**(Forward)**	0.338189	0.629223	0.574764	0.267215	0.084119	0.344554
	(Reverse)	0.519702	0.857965	0.984155	0.422245	0.084119	0.431439
Test of serial	**P-v1**	1.000000	FAILURE	1.000000	FAILURE	1.000000	1.000000
	P-v2	1.000000	FAILURE	1.000000	FAILURE	1.000000	1.000000

3. Software Performance Comparison

Many researchers have conducted tests utilizing various parameters such as area gate equivalent (GE), block size, size of keys, number of rounds, structure, and weaknesses to assess the performance of common lightweight cryptography algorithms. This study reduced the numbers of GE by replacing the complex operation of MixColumns with initial permutation. As well as this study used dynamic numbers of rounds to reduce required resources while keeping the security maintained. Table IV illustrates that the proposed approach achieves better results than the others techniques. It utilized 1280, 1024, and 768 GE with 6, 8, or 10 rounds respectively. However, the proposed approach utilized a dynamic round, so the average number of GE was approximately 1000 GE as shown in *Fig. 5*. These numbers of GE are considered highly efficient with the IoT environment.

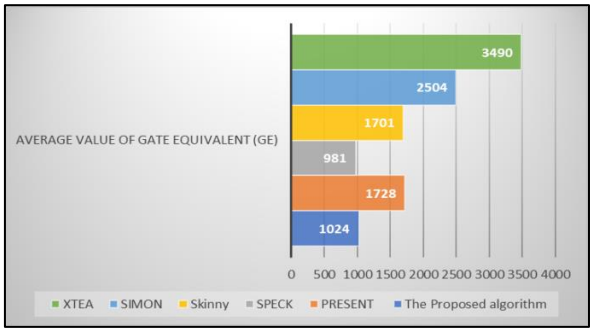


FIG. 5. GATE EQUIVALENT (GE) FOR THE PROPOSED ALGORITHM AND VARIOUS LIGHTWEIGHT ALGORITHMS.

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

TABLE IV. RESULTS OF PROPOSED AND VARIOUS LIGHTWEIGHT BLOCK CIPHERS

Lightweight algorithms	Gate equivalent (GE)	block size (bits)	Size of keys (bits)	Number of rounds	Structure	Weaknesses
The Proposed algorithm	About 1000	128	128	Dynamic (6 or 8 or 10)	Substitution Initial Permutation Network (SIPN)	Understudy
PRESENT 2007[1]	1570 or 1886	64	80 or 128	31	SPN	On the 17th round, related-key attacks, side-channel attacks [22]
SPECK 2013[7]	763, 838, 1000, 984, 1317	32, 48, 64, 96, 128	64, 72/96, 96/128, 96/144, 128/192/256	22, 22/23, 26/27, 28/29, 32/33/34	Add-Rotate-Xor (ARX)	Differential fault analysis, reduced version attacks [22]
SKINNY-64 2016[6]	1223, 1696, 2183	64	64/128/ 192	32/36/40	SPN	On the 11th round, validates the MILP method's efficacy and related differential analysis [28]
SKINNY-128 2016[6]	2391, 3312, 4268	128	128/256/384	40/48/56	SPN	On the 11th round, validates the MILP method's efficacy and related differential analysis [28]
SIMON 2013[6]	1751, 2342, 3419	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192, 256	32, 36, 42, 44, 52, 54, 68, 69, 72	Feistel Network (FN)	Differential fault analysis, reduced version attacks [22]
XTEA 2008[5]	3490	64	128	64	FN	On the 36th round, related key rectangle attacks [22]

4. Throughput Evaluation

Throughput refers to the number of bits created per second at a particular frequency during the encryption/decryption operations. The frequency is either 100 kHz or 4 MHz in the case of hardware or software implementation respectively [22]. The proposed lightweight AES encryption algorithm uses fewer rounds, resulting in higher throughput, faster, and simpler key scheduling. Furthermore, the suggested algorithm's throughput was measured using equation (5) below [29]:

Throughput = Text Size / Time of Encryption (5)

As illustrated in *Fig. 6*, the proposed algorithm has a higher throughput (458.37) than the other comparative algorithms [1] [30]. Therefore, the proposed approach is considered efficient through its high productivity.

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

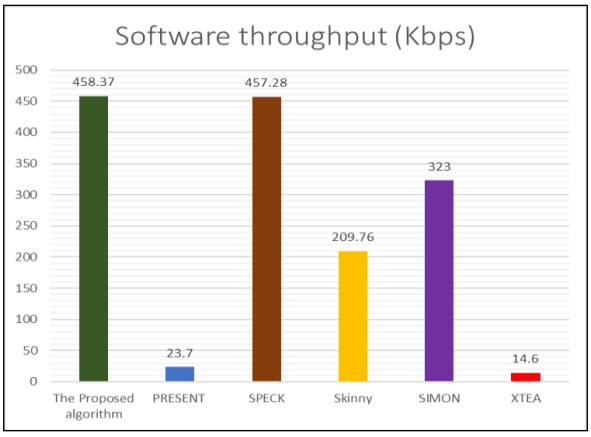


FIG. 6. THROUGHPUT (KBPS) COMPARISON OF SEVERAL ALGORITHMS.

VI. CONCLUSIONS

In recent years, IoT gained major interest from security researchers and academic organizations. Due to it begin used in many fields and it comes to deploying in the real world. Therefore, there are numerous challenges was appeared in IoT ranging from small sensors to servers. As a result, the demand for lightweight encryption methods is growing. So, this paper introduced a Lightweight Advanced Encryption Standard (LAES) depending on dynamic ShiftRows, initial permutation instead of MixColumns, and a dynamic number of rounds. The proposed approach was assessed by using various measurements such as lengths of the key used was 2128 and it is quite enough for security, key sensitivity values were 100%, Also, the proposed approach was compared to XTEA, SIMON, Skinny, SPECK, and PRESENT. The encryption/decryption time of the proposed approach was had the shortest period (0.0169 S) while the SPECK algorithm was had the longest period (4.1249 S) among the comparative algorithms. Whereas, NIST statistical test values of the proposed approach were passed successfully and had higher values than the comparative algorithms. Moreover, the proposed approach utilized 1280, 1024, and 768 GE with 6, 8, or 10 rounds respectively. The average number of GE was approximately 1000 GE. These numbers of GE are considered highly efficient with the IoT environment. In future work, the proposed approach is adapting to deal with higher numbers of data such as the images and assessing them by using image measurements.

CONFLICT OF INTEREST

There are no conflicts of interest declared by the authors.

REFERENCES

[1] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for resource-constrained IoT devices: A Review, Comparison and Research Opportunities," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3052867.

[2] S. Charmonman and P. Mongkhonvanit, "'Internet of Things in E-business," in *Proceeding of the 10th International Conference on e-Business King Mongkut's University of Technology Thonburi*, 2015.

[3] K. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography (NISTIR8114)," *Natl. Inst. Stand. Technol.*, 2017.

[4] M. S. Asaad and M. S. Croock, "Developed Authentication Method for Wireless Sensor Networks Based on Lightweight Protocol," *IRAQI J. Comput. Commun. Control Syst. Eng.*,vol.20, no.4,p48, 2020.

[5] J.-P. Kaps, "Chai-tea, cryptographic hardware implementations of xtea," in *International Conference on Cryptology in India*, 2008, pp. 363–375.

[6] C. Beierle *et al.*, "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Annual International Cryptology Conference*, 2016, pp. 123–153.

[7] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The Simon and Speck

Received 18/October/2021; Accepted 20/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.9

families of lightweight block ciphers cryptology eprint archive." Report, 2013.
[8] A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," in *International workshop on cryptographic hardware and embedded systems*, 2007, pp. 450–466.
[9] J. R. Naif, G. H. Abdul-Majeed, and A. K. Farhan, "Secure IOT System Based on Chaos-Modified Lightweight AES," 2019, doi: 10.1109/ICOASE.2019.8723807.
[10] S. T. Farajaa, S. A. Jassimab, and K. K. Kifayatb, "Mediated IBC-Based Management System of Identity and Access in Cloud Computing," *Tikrit J. Eng. Sci.*, vol. 20, no. 3, pp. 1–9, 2013.
[11] A. Gupta and M. Jaiswal, "An enhanced AES algorithm using cascading method on 400 bits key size used in enhancing the safety of next generation internet of things (IOT)," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 422–427.
[12] S. Madhavapandian and P. MaruthuPandi, "FPGA implementation of highly scalable AES algorithm using modified mix column with gate replacement technique for security application in TCP/IP," *Microprocess. Microsyst.*, vol. 73, p. 102972, 2020.
[13] H. Natiq, N. M. G. Al-Saidi, M. R. M. Said, and A. Kilicman, "A new hyperchaotic map and its application for image encryption," *Eur. Phys. J. Plus*, vol. 133, no. 1, pp. 1–14, 2018.
[14] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, "Designing Substitution Box Based on the 1D Logistic Map Chaotic System," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1076, no. 1, p. 12041.
[15] F. A. Kadhim and M. H. Emad, "Mouse movement with 3D chaotic logistic maps to generate random numbers," *Diyala J. Pure Sci.*, vol. 13, no. 3-part 2, pp. 24–39, 2017.
[16] Y.-Q. Zhang, J.-L. Hao, and X.-Y. Wang, "An efficient image encryption scheme based on S-boxes and fractional-order differential logistic map," *IEEE Access*, vol. 8, pp. 54175–54188, 2020.
[17] M. Z. Abdullah and Z. J. Khaleefah, "Design a Hybrid Cryptosystem Based Chaos and Sharing for Digital Audio," *Iraqi J. Comput. Commun. Control Syst. Eng.*, 2017.
[18] D. Herbadji, A. Belmeguenai, N. Derouiche, and H. Liu, "Colour image encryption scheme based on enhanced quadratic chaotic map," *IET Image Process.*, vol. 14, no. 1, pp. 40–52, 2019.
[19] S. Barshandeh and M. Haghzadeh, "A new hybrid chaotic atom search optimization based on tree-seed algorithm and Levy flight for solving optimization problems," *Eng. Comput.*, pp. 1–44, 2020.
[20] A. S. Albahri *et al.*, "IoT-based telemedicine for disease prevention and health promotion: State-of-the-Art," *J. Netw. Comput. Appl.*, vol. 173, p. 102873, 2021.
[21] S. Jassim and W. Kareem, "Searching over encrypted shared data via cloud data storage," *J. Theor. Appl. Inf. Technol.*, vol. 96, Jun. 2018.
[22] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight Cryptography: A Solution to Secure IoT," *Wirel. Pers. Commun.*, 2020, doi: 10.1007/s11277-020-07134-3.
[23] S. A. Jassim and A. K. Farhan, "A Survey on Stream Ciphers for Constrained Environments," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, 2021, pp. 228–233.
[24] R. Ciesla, *Encryption for Organizations and Individuals: Basics of Contemporary and Quantum Cryptography*. Apress, 2020.
[25] S. A. Banday, M. K. Pandit, and A. R. Khan, "Securing Medical Images via a Texture and Chaotic Key Framework," in *Multimedia Security*, Springer, 2021, pp. 3–24.
[26] T. T. K. Hue, T. M. Hoang, H. X. Thanh, and A. Braeken, "Bit Plane Decomposing Image Encryption Based on Discrete Cat-Hadamard map," in *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, 2018, pp. 344–349.
[27] A. Rukhin, Soto, Nechvatal, Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-allen and hamilton inc mclean va, 2001.
[28] P. Zhang and W. Zhang, "Differential cryptanalysis on block cipher skinny with MILP program," *Secur. Commun. Networks*, vol. 2018, 2018.
[29] M. Saad, "Designing a Secure Environment for IoT Networks Using Lightweight AES Algorithm," *Iraqi J. Sci.*, pp. 2759–2770, 2021.
[30] S. POLAT, "Performance Evaluation of Lightweight Cryptographic Algorithms for Internet of Things Security." Ankara, Turkey: Middle East Technical Univ, 2019.

Received 18/October/2021; Accepted 20/November/2021
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*