Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp.** 46–58
DOI: https://doi.org/10.58496/MJCS/2024/015; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

A New Lightweight Cryptosystem for IoT in Smart City Environments

Firas Hazzaa[1,2,*], Md Mahmudul Hasan[1], Akram Qashou[1], Sufian Yousef[1]

[1] *Faculty of Science and Engineering, Anglia Ruskin University, Cambridge, UK*

[2] *Higher Education & Scientific Research ministry, Iraq*

ARTICLE INFO

Article History

Received 13 Jul 2024
Accepted 11 Sep 2024
Published 13 Oct 2024

Keywords

Internet of Things IoT

Secure Communication

Lightweight Cryptograph

AES encryption

Security

cc BY

ABSTRACT

Internet of Things (IoT) devices, user interfaces (UI), software, as well as communication networks are all deployed within Smart Cities topology. The security approach designed for Internet of Things IoT should be able to prevent and detect both internal and external attacks. The problem in IoT network that not every linked node or device has an adequate amount of processing power. This means that data encryption and other related activities will be impossible and means that the security of any kind must be lightweight. A trustworthy security solution that stops illegal access to private data on the network is necessary for maintaining the privacy of information on the Internet of Things. Cryptographic processes need to be quicker and more compact without sacrificing security. The aim of this study is to reduce the execution time and power consumption of encryption processes without compromise the complexity of the encryption algorithm. This research presents a new lightweight cryptographic technique to protect various multimedia and real-time traffics across IoT network, by using two S-box in SubByte of encryption process, without affecting its performance. In this study, different audio samples will be used to test the new algorithm efficiency. Comparing the suggested method to the most advanced standard algorithm, it can reduce the cryptography process's execution time as well as energy consumption while maintaining the required security level. The outcomes demonstrate good performance in terms of power usage and delay. The new technique consumed a roughly 0.2 µJ for encryption process while the typical AES algorithm consumed 0.29 µJ, this mean the new algorithm achieved (33% power savings), while maintaining a good complexity level (security) within the process of encryption according to the results in tables I, II, and the comparison in table III. The novelty of this work can be showed by using dual XOR S-box technique which increased the complexity of SubByte process making it more secure without overload the processing performance, in addition to the reduction in encryption rounds which contribute to enhance the performance without compromise the security. Making it more suited for the Internet of Things (IoT) used in smart city environments.

1. INTRODUCTION

Smart cities use their networks of interconnected IoT smart nodes/sensors/wires and other systems to enhance the population’s living conditions and economic growth. The IoT is a collection of objects, such as sensors or home-use appliances, which are connected and can transfer information. The info generated and provided by IoT sensors and devices or nodes is stored in the cloud or servers. Integrating these devices and data analysis helps bridge the gap between physical and digital city components, enhancing the efficiency of the public and private sectors in achieving economic opportunities and providing better living standards to citizens. Concerns over the security of connected devices and sensors remain high for stakeholders; as Chin noted, the cost of cybercrime is expected to reach $8 trillion in 2024 [1]. Thus, most companies are paying significant attention to security issues.

Our motivation is to enhance IoT security without affecting its performance. The problem in this network that not all connected sensors have enough power to perform big task such as data encryption. So we did this study to address this issue by develop new encryption algorithm with lightweight features and high security strength. This is because cryptography algorithms have a lot of iterative processes that take time to complete and consume a lot of battery power.

Encryption and security are the two measures that safeguard both communication and info. Security specifies and restricts what can or must go where, when, and how, while cryptography offers means of achieving the above security. Furthermore, audio on wireless IoT networks and, in general, multimedia applications remain topical and influential as they facilitate

[*] *Corresponding author. Email: Firas.hazzaa@pgr.anglia.ac.uk*

human interactions. However, the key issue in the networks is the security of the connections between the nodes in the IoT [2] [3]. A major problem with managing voice applications in wireless IoT networks is finding a security solution that will fit the application requirements, such as delay and low energy. Any enforcement of cryptography has to be efficient and timely. Lightweight refers to the fact that the application does not consume too many resources in terms of energy and battery. Another security principle is confidentiality, which can be implemented using cryptographic techniques. These algorithms have been applied in many applications and methods to address security requirements. Thus encryption/decryption has become an important aspect of security [4].

However, in wireless IoT networks, the problem is that not all connected nodes have the necessary computational resources. This means that tasks such as data encryption will be challenging, and any security must be lightweight [5]. The objectives of this research is to introduce new S-Box subByte process to increase the complexity without effect the performance and to use 9 round iteration to enhance the execution time and energy consumption, in addition to conduct comprehensive security analysis to prove its robustness.

Today, many cryptography algorithms can be employed to protect the data. AES is the most secure and the least vulnerable encryption algorithm among all the four algorithms. AES has been classified as top secret by NIST and is widely used in many applications. The four functions consist of signals such as AddRoundKey, Sub-Byte, Shift-Rows, and MixColumns. The generation of the key is an iterative function, and depending on the key's field size, several steps are performed [6]. Many rounds can be used 14 for key sizes of 128, 192, and 256, respectively. Advanced Encryption Standard subdivides every 128 bits of the plaintext blocks into a state matrix that is a four-by-four bytes matrix. The SubBute function employs only one S-box to substitute for plain text during encryption.

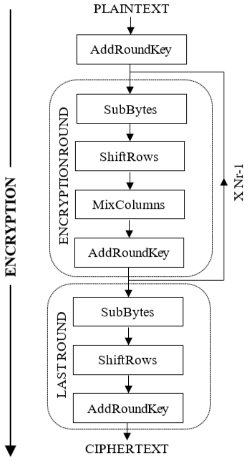


Fig. 1. Standard AES Algorithm

This study shows the strength of development of a new encryption algorithm that enhances the encryption execution time and energy consumption and maintains the desired level of security which achieved optimal tradeoff (balance) between the security and the performance. The changes proposed here imply that this algorithm should be more appropriate for audio traffic over IoT and contribute to the search for the right balance of security and performance. By effectuating the necessary alterations to the execution flow of the AES algorithm processes, its mission is to introduce a new dual S-box-XOR transformation process to boost the security feature or the cryptography and employ other strategies that will help reduce its effect.

The key contribution of this research is to enhance the security and the performance of IoT network. The problem in this network that not all connected sensors have enough power to perform big task such as data encryption. So we did this study to address this issue by develop new encryption algorithm with lightweight features and high security strength. Our solution introduced new S-box XOR method for substitution function used in encryption algorithm to secure the data; while maintaining the performance of the algorithm in acceptable manner which showed excellent tradeoff between security and performance as illustrated in tables III and VII. The additional contribution is conducted a security analysis to test the strength of the new design. The robustness of our proposed encryption algorithm is proven in the security analysis section. The aim of the security analysis used in this research is to confirm the security strength of the proposed schemes as this can be considered as one of the contributions of the paper as there is a lack of security analysis in many research works. The new algorithm has achieved 32% energy saving comparing with state of arts AES algorithm. The paper is organized as following: Section 2 demonstrates the literature review, section 3 illustrates the proposed framework, Section 4 shows the results and security analysis, section 5 explains the conclusions and future work.

2. LITERATURE REVIEW

The research and development of lightweight cryptography was initiated in 2004 in Europe through a project, and it has been revived through the M2M/IoT process. At ISO/IEC JTC 1/SC 27, the international standard ISO/IEC 29192 "Lightweight Cryptography" was created. The Lightweight Cryptography Project essentially was commenced in 2013 by the U.S. NIST (National Institute of Standards and Technology), which publishes standards on cryptographic technologies, [8] stated. In 2017, the NIST made an announcement regarding a public request for applications of lightweight cryptographies. Numerous studies have been carried out to investigate security concerns pertaining to the actual traffic on wireless networks. While a portion of them concentrate on the intricacy pertaining to the security method and disregard the performance requirements (encryption latency, energy), others concentrate on building algorithms that are suitable for coping with real-time limits like delays, speed, and energy saving.

As per research by [2], an adaptable, low-power encryption scheme for Internet of Things devices is recommended. Their aim is to overcome the memory and processing capacity constraints of IoT devices, the system encrypts and decrypts data using a variable data size and robust logical operations. When it comes to encryption time, the suggested encryption method has outperformed AES however this study has lack of comparable security analysis with the current cryptosystem. A low-power as well as lightweight encryption technique is suggested by the authors in [3] for safeguarding voice communication over wireless networks. The suggested algorithm works well with wireless devices and satisfies voice traffic requirements. When compared to the most recent standard algorithm, it can minimise the encryption process's execution time and power consumption while preserving the intended security level. Its considerable time and energy savings, roughly 35% greater than with the regular AES algorithm, combined with a respectable degree of encryption process complexity make it more appropriate for use in wireless environments.

Recent studies on lightweight cryptography for IoT security, such as TinyCrypt [6] and research on resource-constrained devices [7], highlight the importance of optimizing both energy efficiency and security. TinyCrypt, a small footprint cryptographic library, is designed to operate efficiently on low-power IoT devices, offering a range of cryptographic primitives like ECC and SHA-256. Its key strength lies in its ability to provide strong security while minimizing computational and energy overhead. Similarly, the study on Lightweight Cryptography for Resource-Constrained IoT Devices reviews algorithms like SIMON and PRESENT, emphasizing solutions that balance security with resource efficiency. Both approaches address the challenges of constrained environments, aiming to secure IoT systems without overburdening limited resources.

Compared to these methods, the proposed algorithm achieves a 35% energy savings, making it competitive with TinyCrypt in terms of energy efficiency. It also delivers a "same or higher" security level without relying on hardware optimizations, making it suitable for a wide range of IoT devices. Unlike some lightweight cryptographic algorithms that benefit from hardware support, the proposed method is purely software-based, offering a flexible and cost-effective solution. Overall, while TinyCrypt excels in modularity and adaptability, the proposed method remains a strong alternative, particularly for environments that cannot accommodate hardware upgrades.

Abutaha, M., et al. [10] presented a paper in which they offered a lightweight, secure cryptosystem for the Internet of Things and small devices. These devices last longer because of the technology's lower power consumption. Additionally, this system uses less memory, which makes it perfect for IoT and small devices. The study demonstrated a workable hardware lightweight cryptosystem that was constructed with the Verilog programming language on an FPGA board. The recommended hardware solution is predicated on resource consumption suitable for Internet of Things devices and real-time applications. Additionally, this system has demonstrated strong performance at high throughputs, allowing for real-time data transfer. However, the authors are also concerned that there are still a lot of unanswered issues about modules pertaining to hardware security, which present a challenge that is expensive and requires caution. Additionally, authors in [15] propose a 5-round AES encryption technique for wireless sensor network multimedia as well as real-time applications. The execution time was shortened, according to the results. Nonetheless, there is a chance of hacking because cryptanalysis can easily crack five rounds of encryption.

A new lightweight, which is a compact encryption scheme rooted in bit permutation instruction group operation (GRP) essentially was instituted by Bansod et al. in [16]. They suggest a brand-new hybrid system that provides more condensed outcomes for embedded security gate equivalents and memory usage. According to the authors, typical algorithms like DES and AES need a significant amount of memory and cannot be implemented for an embedded system method and edge devices. Nevertheless, their research overlooks the important security analysis that is lacking.

However, due to the multiple sensitive requirements, such as latency, throughput, and energy consumption in IoT environments, none of this literature is acceptable for safeguarding real-time applications essentially over wireless IoT networks. The truth is that security (confidentiality) and performance (encryption latency, energy) are always trade-offs. We believe that the simplest way to tackle these needs is to suggest a new strategy that modifies a few functions in the existing cryptographic algorithms, making it lightweight, appropriate for V-over-IoT, and helpful in achieving a decent trade-off between security along with QoS metrics. From our point of view, our solution is a better choose for IoT network security, because it showed excellent tradeoff (balance) between security and performance, compared with existing studies which lack of acceptable tradeoff which can be clearly seen in table VII at the end of this paper.

3. THE PROPOSED FRAMEWORK

The methodology to investigate the problem of audio cryptography IoT nodes conduct a performance testing against different voice data sizes. It contains the (new SubByte) dual XOR S-box creation for complexity enhancement, and utilization of new mixcoloumn and 9 rounds iteration in new proposed algorithm. The Validation & evaluation of component elements for the building of a framework design that permits new algorithm to be used for crypto the voice over IoT network, Fig. 2 shows the Methodology and illustrations the new algorithm stages while the following sub-sections explain the amended functions.

This study used a quantitative approach and observational investigation as part of its investigatory strategy. The experimental strategy was designed using scientific methodology. To support the design choices for a novel lightweight algorithm, research objectives were established to gather primary evidence from the existing literature along with acquiring empirical proof over experimentation. Such design components were assembled into the prototype, which then was put to the test purposed for seeing how well it performed compared to the industry standard procedure as well as comparable up-to-date research undertaken by other authors. Generally, the answers to research questions were made possible by the study of empirical data gathered via tests. By gathering the numerical data, the quantitative research technique assisted in this study's interpretation of statistical security analysis. It provided a useful comparison analysis with existing algorithms as well.

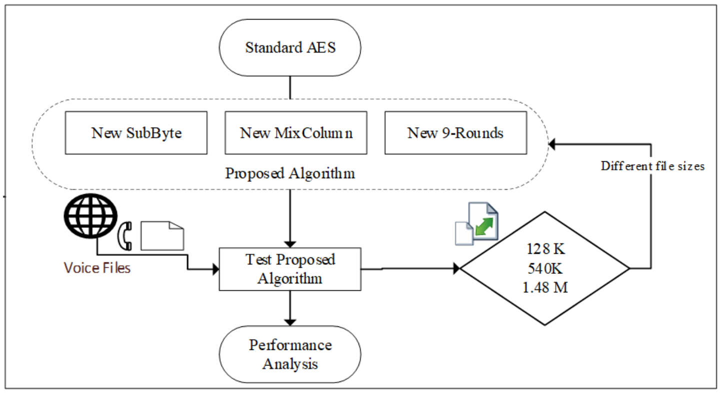


Fig. 2 Methodology & the New Algorithm

3.1 New XOR S-box Procedure

Encryption/decrypting functions cipher and inv_cipher using the substitution tables (s_box) and (inv_s_box) to directly substitute a byte GF(2^8) by another byte of the same finite field.

Refer to Fig.3a. find_inverse is the first step in the S-box generating process is to search for the multiplicative inverses of all elements of the finite field GF(2^8). Or, for all possible 256-byte values b, find the byte b−1 that satisfies

$$b*b^{-1} = 1 \qquad \textit{Where * represent a polynomial multiplication defined in poly\_mult.}$$

After that, the second stage is an affine transformation, which involving of a polynomial multiplication with a specific constant ($31_d = 00011111_b$) modulo another constant ($257_d = 100000001_b$) and the XOR addition of a third constant ($99_d = 01100011_b$):

$$b_{out} = b_{in} \bullet 31_d \bmod 257_d + 99_d \qquad \textit{where + bit-wise XoR operation.}$$

AS explained, AES algorithm using **one** S-box for subByte transformation process which may vulnerable to some attacks. So the suggested technique in this research is to generate **two** substitution S-boxes to increase the complexity of the crypto-system. and then XOR these two S-boxes to create the new S-box that using in the substitution method of SubByte process of the new algorithm. This technique requires just two keys K and C to generate (S-box1 and S-box2) which are then summed (XOR) to produce the new S-box which will be utilized in the SubByte process, as illustrated in Fig.3. We are not increased more S-boxes here, as we did in our previous work [3], to avoid memory limitation or overload the encryption process.

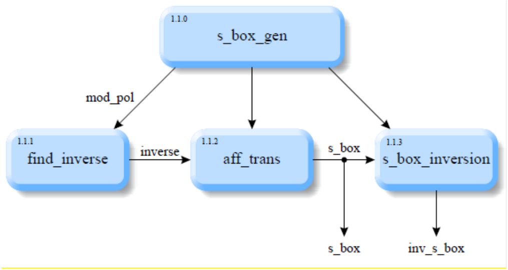


Fig. 3a. S-Box generation

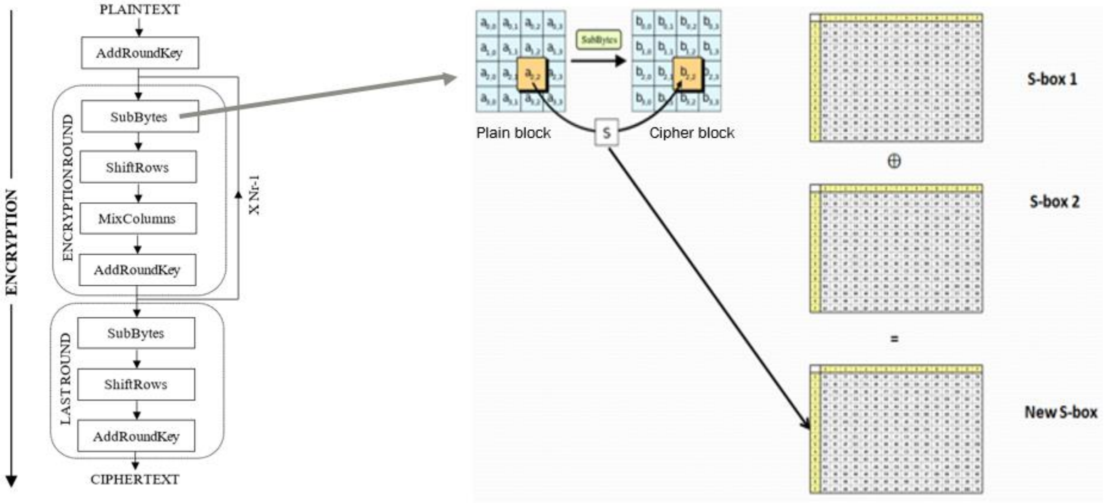


Fig. 3 XOR S-Box 1 & 2 to produce new S-box

The approach will decrease the keys creation number and generate just one additional S-box. Therefor the generation of many-S-boxes not required. This technique enhances the effect of wasting several S-boxes on the memory size of the processing unit and substitution process. There are (2^{16}) possible S-boxes will be created through the new systems employing two k only (8 bit for each key) according to the following:

$$complexity = 2^8 * 2^8 = 2^{16}$$

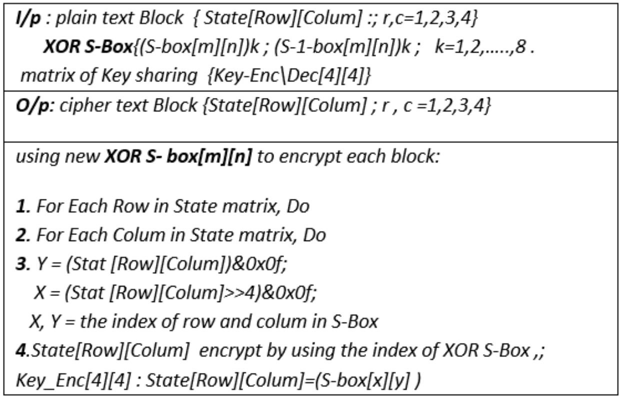
Algorithm 1 (proposed S-Box creation with XOR operation)

```
I/P: Randomly (2) values as { Rndm_Key[k], Cons_c[k] , k=1,2}

O/P:  S-Boxes { (S-box[m][n])k ; (S-1-box[m][n])k}

1. Choice 2 keys Rndm_Key[k]
2. Choice 2 values Cons_c[k]
3. For each key Rndm_Key[k]  & relates constant Cons_c[k] generate its own S-box[m][n]
(S-box[m][n])k= Rndm_Key[k] * mulp[r][c] + Cons_c[k]
Where mulp[r][c] represent the multiplicative inverse in GF($2^8$)
4. XOR the two (S-box[m][n]) k.
5. Get the new S-box and used in the encryption.
```

Algorithm 2 Encryption with XOR S-box



```
I/p : plain text Block  { State[Row][Colum] :; r,c=1,2,3,4}
     XOR S-Box{(S-box[m][n])k ; (S-1-box[m][n])k ;   k=1,2,.....,8 .
 matrix of Key sharing  {Key-Enc\Dec[4][4]}
O/p: cipher text Block {State[Row][Colum] ; r , c =1,2,3,4}

using new XOR S- box[m][n] to encrypt each block:

1. For Each Row in State matrix, Do
2. For Each Colum in State matrix, Do
3. Y = (Stat [Row][Colum])&0x0f;
   X = (Stat [Row][Colum]>>4)&0x0f;
  X, Y = the index of row and colum in S-Box
4.State[Row][Colum]  encrypt by using the index of XOR S-Box ,;
Key_Enc[4][4] : State[Row][Colum]=(S-box[x][y] )
```

3.2 Utilise MixColumn process and New 9-Rounds Iteration

This section provides an implementation of the previous work in our research [3]. As we explained in the last sections, the AES has the MixColumn process, and is considered one of the most time-consuming because the input matrix is multiplied over GF (2^8) [3]. As was described in Section IV of our prior work [3], the modified mixcolumn function can reduce a vast amount of time for the execution, which consequently contributes to enhancing the effectiveness of the encryption process. In addition to this new proposed enhancement of this work, the reader may find [3], page 5 – section 3. 3. 2 for more details on the role of the mix column function.

Another modification to our suggested algorithm is reducing the encryption and decryption rounds by incorporating a New nine-round iteration. It, therefore, means that if we reduce the number of rounds in AES, then it is faster and consumes less power; as we pointed out earlier, where we discussed in section (1), the complex Mix Sub-Byte transformation that may reduce the number of rounds of AES that will be appropriate for IoT devices with limited power nodes and detailing the New Nine-Rounds Iteration. The wide explanation is explained in reference [3]. Similarly, the security level remains the same due to the algorithm’s complexity being overcome in generating the S-Box, not in a round.

3.3 The New Lightweight Algorithm

This section shows the overall design which introduces additional functionalities to the AES algorithm outlined earlier, achieving a dependable balance between security and (QoS) metrics. This enhancement leads to decreased power and time consumption while bolstering the algorithm's security. The following formula presents the expanded overall key space of the revised algorithm.

$$Dual\ Key\ (\ new\ key\ space):\ \ 2^{128} * 2^{256} * 8!$$

For attackers to breach the system, they must simultaneously possess knowledge of two keys, rendering the task significantly challenging since guessing both keys concurrently are highly improbable. Thus, even if one key is compromised, locating the other remains a formidable challenge. Moreover, in theory, employing a Brute-force attack to uncover all keys would demand an immense amount of time, exceeding 5.3*10^23 years for success.

Consequently, the security level of this algorithm remains intact or possibly even enhanced under such circumstances. As previously indicated, the updated algorithm is better suited for wireless IoT environments due to its novel features. The subsequent section offers further analysis connected to these enhancements. Figure 4 illustrates the suggested improvements in the revised AES algorithm. According to the fig. (1) and (4) there are clear differences between the two algorithms, for example the new XOR subByte function is clearly appeared in fig.4.

To validate and evaluate the new algorithm, a comparison with the standard AES algorithm has been carried out to compare the security strength parameters such as the randomness. These parameters can be measured by running various types of tests such as binary histogram and entropy which explained in the security analysis section. In addition, the time and power consumption are compared.

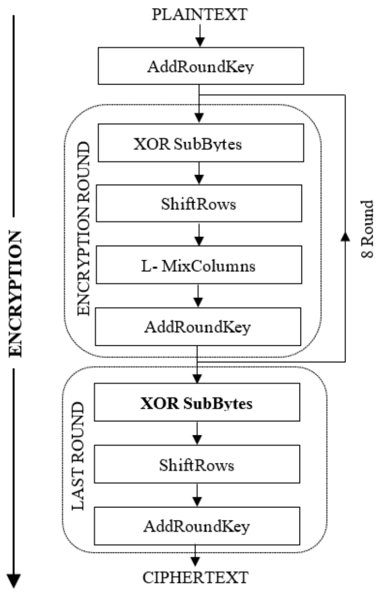


Fig. 4 The New Algorithm

The testing has been conducted in a laboratory environment using a High-performance lab computer with Microprocessor (quad-core i7, 8 GB RAM) running Windows 7, running Microsoft Visual Studio 2015 to build and write the proposed algorithm code using visual C++ programing language, which executes the encryption/decryption process. Console app used to avoid any load on the processor and avoided additional delays. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft, for more accuracy, the code has been run for five times and the average output has been calculated. The experiments and testing will be carried out on Audio file with .wav format and in different sizes, 128 KB, 540 KB, 1 M, and 1.48 MB and another size. The parameters measured are, Delay and energy, and for security analysis are, Binary Histogram, Randomness *P-value,* and Entropy. The key used in this experiment is *123456789abcdef123456789abcdef12*. The length of the key is 32 char (i.e. 16 byte).

4. RESULTS

The findings of our experiments are shown in this section. The time and energy used for encryption and decryption by the standard AES method as well as the recommended approach are displayed in Tables 1 and 2.

TABLE I. DELAYS

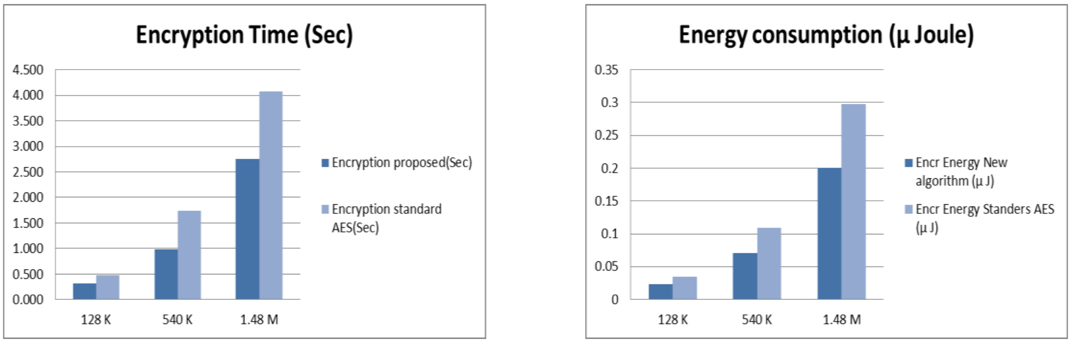
Audio file	Encrypt (Sec) (AES)	Decrypt. (Sec) (AES)	Encrypt. (Sec) (New Lightweight)	Decrypt. (Sec) (New Lightweight)
128 KB	0.4765	0.461	0.3141	0.3242
540 KB	1.7421	1.385	0.9772	1.0861
1.48 MB	4.0774	3.884	2.7471	2.8331

TABLE II. ENERGY CONSUMED

Audio file	Encrypt Energy µJ (AES)	Decrypt Energy µJ (AES)	Encrypt Energy- µJ (New Lightweight)	Decrypt Energy- µJ (New Lightweight)
128 KB	0.0349	0.0341	0.0229	0.0242
540 KB	0.1088	0.1012	0.0712	0.0791
1.48 MB	0.2979	0.2841	0.20	0.2069

For all file sizes, the AES algorithm consistently takes longer to both encrypt and decrypt files compared to the New Lightweight algorithm. The time increases significantly with file size, showing a more substantial increase as the file size grows. The New Lightweight algorithm is faster than AES for both encryption and decryption across all file sizes. Although the time also increases with file size, the increase is less pronounced compared to AES, suggesting better scalability. For example, the encryption of 128 KB File using AES takes 0.4765 seconds, and decryption takes 0.461 seconds. While the

New Lightweight Encryption takes 0.3141 seconds, and decryption takes 0.3242 seconds. The New Lightweight algorithm is approximately 34% faster for encryption and 30% faster for decryption compared to AES. For 1.48 MB File**,** AES Encryption takes 4.0774 seconds, and decryption takes 3.884 seconds. While the New Lightweight Encryption takes 2.7471 seconds, and decryption takes 2.8331 seconds. The suggested algorithm's outcomes demonstrate notable advancements in the encryption and decryption procedures. The execution time is improved by more than 33% for a range of file sizes. The suggested approach, for instance, improves the encryption time by about 33% for a file with a size of 1.48M. Similar behavior is shown in the metric for energy consumption, where the suggested algorithm outperforms the standard AES by more than 32% (see fig. 5). This indicates that the method improved the subByte function and used the same quantity of time as well as energy as in [3].



(a) (b)

Fig. 5. Difference in (a) encrypt time (b) Energy consumed by new algorithm and AES

The properties of the suggested algorithm and the standard AES are contrasted in Table 3. It is evident that they differ significantly from one another. Furthermore, it is evident that the file size has not changed, meaning that neither the memory capacity nor the network path's bandwidth would be impacted.

TABLE III. COMPARISON OF FEATURES BETWEEN THE STANDARD AES AND THE PROPOSED ALGORITHM

Metrics	**AES**	**New Algorithm**
Function	Scure	Scure
S_box	Single	Two XOR
No. of keys	1	2
Crypto Key space	2^{128}	$2^{128}*2^{16}$
B - Histogram	Random	Random
Rounds	10	9
Block size	16 Bytes	same
Power consumed	**P**	**0.68 P**
Crypto time	**T**	**0.67 T**

S-box creation is a pre-process step done before the encryption, so it does not impact performance during the actual encryption process. This allows the use of a more complex S-box (the new XOR S-box) in the encryption phase. Therefor we are successfully used two S-box in the new algorithm, (means more complexity in the encryption process), and at the same time this algorithm consumed less amount of time and energy because of using 9 rounds iteration., as shown in the results above. Table 3 indicates that each key has a complexity of 2^{128}. Moreover, the use of two S-Boxes raised the difficulty. The complexity was then increased by the number of rounds, with the normal AES having 10 rounds and the suggested alteration having 9 rounds. In the meantime, the new MixColumn function and fewer rounds have resulted in a 33% saving in execution time and energy usage. The New Lightweight algorithm achieves similar security levels with fewer rounds. This reduction reduces the number of transformations applied to the data, thereby reducing both the encryption and decryption time. Fewer rounds also mean less energy consumption; as less computational resources are required for each operation. The suggested algorithm's security strength is evaluated and validated by comparing its security metrics with those of the conventional AES in detail in the next section.

4.1 Security Analysis

The security metrics are crucial for determining the level of security in any security system, claims [31]. Numerous security metrics, including entropy and the Binary Histogram test, have been measured. Each test was run on an audio file which has been encrypted. *The Numerical Assessment Suite (P-RNG)* performs additional security analysis on the voice data both pre & afterward the suggested LEA technique has encrypted them. The National Institute of Standards and Technology's Statistical Test Suite is a valuable tool for analyzing random and pseudorandom number generators in the context of cryptographic applications [30]. In this research, the encrypted data randomness has been tested using the frequency test. The investigation calculates the P-value of every bit stream that it has selected. It also computes the result by counting the ones and zeros in each stream. One can compute the P_value by applying the threshold (alpha 0.01). The test is considered successful when the bit streams produce results higher than alpha.

The following steps outline the computation of the P_value:

The input sequence (e), consisting of zeros and ones, is transformed by converting zeros to -1 and ones to +1. These converted values are then summed to yield S_n. For instance, if

$e = 1011010101$, then n=10 and $S_n = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2$

Compute the test statistic.

$S_{obs} = \frac{|Sn|}{\sqrt{n}}$, then

$S_{obs} = .632455$

$P\text{-}value = \boldsymbol{erfc}\,\frac{|Sobs|}{\sqrt{2}}$, where ***erfc*** is the complementary error function.

If the $P\_value \geq 0.01$, then the sequence is random. Otherwise, conclude it non-random.

TABLE IV. (ORGINAL VOICE)

P-Value of selected Bit stream & No. of Zero's and one's of Bit stream

Streams	.P- value	Outcome	measure	Bitstream	0s	1s
1st	0.0000041	Failed	0/5	100	74	26
2nd	0.0000629	Failed		100	71	29
3rd	0.0000021	Failed		100	75	25
4th	0.0000028	Failed		100	73	27
5th	0.0000022	Failed		100	76	24

The tables presented above display the P-values and outcomes of bit stream. Five-bit streams were selected for this test based on our specified instructions.

TABLE V. (CIPHER VOICE)

P_Value for selected streams & No. of Zero's and one's for each Bitstream

Bitstream	P_value	outcome	measure	Bitstream	0s	1s
1st	0.20761	Passed	4/5	100	57	43
2nd	0.01629	Passed		100	58	42
3rd	0.10522	Passed		100	46	54
4th	0.00080	Failed		100	61	39
5th	0.42321	Passed		100	43	57

The encryption algorithm exhibits significant randomness, successfully passing 4 out of 5 randomness tests. Additionally, there is a nearly equal distribution of zeros and ones in the output.

4.1.1 Entropy Result

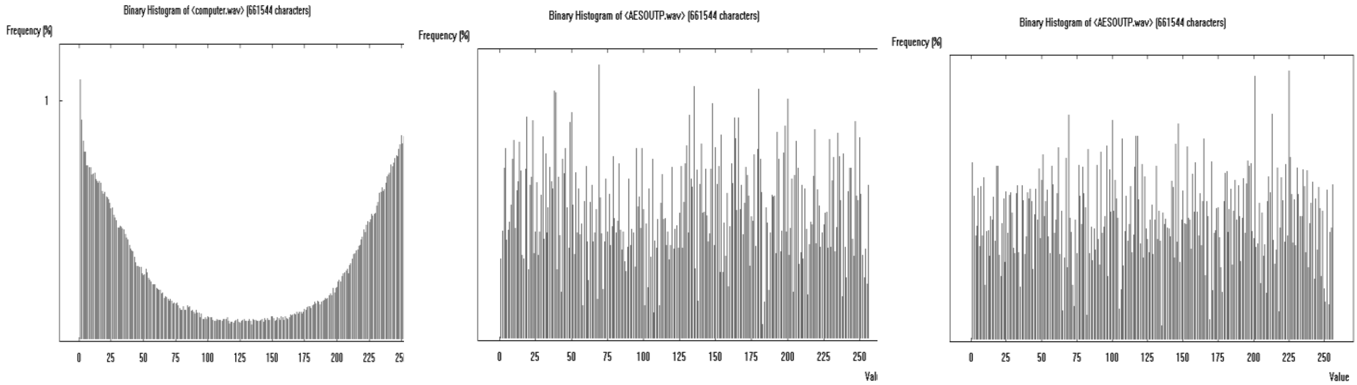
An index of a document's content of data is its entropy. Basically, the bits per character represent the entropy. In this sense, the results of the Entropy test for the file which has been encrypted using the conventional AES and the suggested technique are displayed in Table VI. When compared to the usual algorithm, which yielded a score of 7.99 out of a possible 8, the new algorithm obtains the same score. This indicates that the suggested algorithm sustains the same security level.

TABLE VI. ENTROPY ANALYSIS

Audio file size / Byte	before	AES algorithm			New algorithm		
	Entropy	Entropy	Max. possible Entorpy	Possible byte value	Entropy	Max. possible Entorpy	Possible byte value
540k	7.79	7.99	8	256	7.99	8	256

4.1.2 Binary Histogram

The binary histogram visually represents how often each character in the document occurs. The following illustrations display the binary histogram for the processed voice, comparing the outcomes between AES algorithm and the newly suggested algorithm. Figure 6a depicts the Binary Histogram for initial voice prior to cryptography.



(a) (b) (c)

Fig. 6. (a) the Binary Histogram of original voice, (b) the Binary Histogram of encrypted voice using (proposed algorithm), (c) the Binary Histogram of encrypted voice using (AES)

Figure 6b illustrates the Binary Histogram of the audio file using the new encryption algorithm. A high level of protection is evident when comparing the first audio file to the latest output from the algorithm. The Binary Histograms of the encrypted file through the standard AES and the Binary Histogram being incorporated in the new algorithm are shown in Figure 6c. This means that there are no changes in the degree of difficulty of the algorithm; hence, the security (complexity) level has been attained.

The preceding analysis and tests unequivocally demonstrate that the suggested technique attains a substantial level of security. Its robust resistance to various attacks stems from its inherent characteristics of confusion and diffusion. Moreover, the significant randomness observed in the preceding subsections indicates that the recommendation of a 9-round iteration did not compromise the security level. Parameters for analyzing the security level, such as the binary histogram, were compared with those of the standard AES, revealing equivalent security strength.

4.2 Evaluation

In addition to the comparison, which been done in table III, between our proposed algorithm and AES algorithm, Table VII provides a comparison between this research and previously conducted studies. It contrasts the energy saving percentage and the lightweight security level achieved relative to the state of art AES algorithm.

TABLE VII. SAVED ENERGY RATIO AND SECURITY LEVEL.

Pre. Work	Saved Energy %	Security level
[2]	Less energy	No comparable analysis
[3]	35%	High / 8 S-box
[15]	41 %	- Low/ 5 rounds
[10]	Less power	High/ hardware restrictions
Proposed algorithm	33 %	Same/high /no hardware used

The table shows a comparative analysis of several studies based on saved energy ratio, and security level. The algorithm [2] uses less energy but lacks a detailed analysis of its pre-work and security features. Where the suggested algorithm in our previous work [3] achieved a very good energy saving with high level of security however it uses 8 S-boxes which may affect the memory space. Also, the work [15] achieving a notable 41% energy savings but operates at a low security level with 5 rounds of processing. Furthermore, the research [10] is also efficient in terms of power usage and is characterized by a high security level but enforced through hardware restrictions. Where our proposed algorithm stands out by saving 33% energy and providing the same or higher security without the need for many more S-boxes or hardware involvement, in addition to a comprehensive comparison against AES algorithm has been presented in our work, indicating robust software-based security mechanisms. Based on the data presented in the table, the proposed algorithm has successfully achieved a notable trade-off solution between security and Quality of Service (QoS) metrics. This makes it particularly well-suited for voice traffic across IoT with constrained resources.

5. LIMITATIONS

Our research demonstrates an encryption technique for voice data over IoT network, however it has some limitations. The proposed scheme, which employs dual XOR S-boxes and resilient adjustments to AES functions, aims to improve lightweight security while optimizing delays and energy consumption. Despite its benefits, the technique’s efficiency is demonstrated primarily with voice samples and may not generalize to other data types. Moreover, while the new algorithm decreases energy usage by up to 33%, it is tailored to specific IoT environments like smart cities. Further research is desirable to measure its applicability and performance through different IoT scenarios and data forms.

6. CONCLUSIONS

In this research, a novel cryptography technique pertaining to voice across IoT network is suggested, verified, as well as tested. The aim is to rise the lightweight security of the encryption algorithm by using dual S-boxes while maintaining the optimal performance in terms of delay and power consumption during the cryptography procedure. The new algorithm can match the performance requirements of such devices/nodes as well as networks and offer a decent level of security by utilising strong modifications to the basic AES functions. The Sub-Byte process has been modified to raise the level of complexity during the encryption and decryption procedures while maintaining the same delay time and memory size. An optimal utilisation of the MixColumn function and nine rounds iteration was involved in this design, when less time and energy are required to perform the process of the encryption. The proposed system can minimise the execution time by almost the same amount while reducing the power usage by up to 33%. Which explained in tables I, II, and the comparison in table III The paper's primary contribution is introduced new S-box XOR method for substitution function used in encryption algorithm to secure the data and address the memory size limitation when compared to the conventional AES algorithm. A security analysis has also been carried out to test the strength of the new crypto system. The suggested algorithm demonstrates its usefulness for any wireless IoT network with constrained resources since it can balance quality of service and security in a significant approach. It is especially appropriate for wireless IoT devices/nodes, having restricted resources, deployed in smart cities.

For future work, we plan to investigate other aspects of AES, such as the ShiftRows and AddRoundKey operations, for example, utilizing Triple keys will strengthen the algorithm security. This broader exploration could potentially yield even more comprehensive improvements in terms of performance, security, and energy efficiency. In addition to further exploration of additional file sizes is planned to be tested to capture the performance and scalability of the algorithm for much smaller or larger data sizes, which could be crucial in IoT systems with highly variable data payloads. Furthermore, we can consider to test many file formats and different file types such as video or text file. By expanding the scope of modifications, we aim to push the boundaries of the current lightweight encryption paradigm and discover additional opportunities for innovation.

Conflicts of Interest

"The authors declare no conflicts of interest".

Funding

Not Applicable.

Data availability: we used our dataset and its available on request.

Acknowledgment

“The authors thank the Higher Education and Scientific Research ministry in Iraq for support and fund this research in addition to thank the anonymous reviewers forgiving their valuable comments and helping us to improve the quality of the paper.”

References

[1] S. Morgan, "Cyber Security Cost," Forbes, available online on: https://www.forbes.com/sites/stevemorgan, US, 2022.

[2] M.A.F. Al-Husainy, B. Al-Shargabi, S. Aljawarneh “Lightweight cryptography system for IoT devices using DNA” Comput. Electr. Eng., 95 (2021), Article 107418.

[3] Firas Hazzaa, Antesar M. Shabut, Nada Hussein M. Ali, Marcian Cirstea, Security Scheme Enhancement for Voice over Wireless Networks, Journal of Information Security and Applications,Volume 58, (2021) , ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2021.102798.

[4] M. Masoumi and M. H. Rezayati, "Novel Approach to Protect Advanced Encryption Standard Algorithm Implementation Against Differential Electromagnetic and Power Analysis," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 10, no. 2, pp. 256-264. [available online]: (www.ieeexplore.ieee.org), Digital Library, February 2015.

[5] L. S. Abhiram, B. K. Sriroop and H. L. Punith.Kumar , "FPGA implementation of dual key based AES encryption with key Based S- Box generation," India, 2015.

[6] Y. Li, J. Liu, and Y. Jiang. "TinyCrypt: A Small Footprint Cryptographic Library for Low-Power IoT Devices." IEEE Internet of Things Journal, vol. 6, no. 1, pp. 768-779, Feb. 2019. DOI: 10.1109/JIOT.2018.2876226

[7] M. Ali, T. Javed, F. A. Sheikh, and S. Aslam. "Lightweight Cryptography for Resource-Constrained IoT Devices: A Review and Future Perspectives." IEEE Access, vol. 9, pp. 167471-167497, Nov. 2021.

[8] OKAMURA Toshihiko, Lightweight Cryptography Applicable to Various IoT Devices, Vol.12 (2017) No. 1 Special Issue on IoT That Supports Digital Businesses /www.nec.com/ .

[9] J. Wang, Q. Gao, P. Cheng, Y. Yu, “Lightweight Robust Device-Free Localization in Wireless Networks," *IEEE Transactions on Industrial Electronics, IES,* vol. 61, no. 10, 2014.

[10] Abutaha, M., Atawneh, B., Hammouri, L. et al. Secure lightweight cryptosystem for IoT and pervasive computing. Sci Rep 12, 19649 (2022).

[11] H. Albonda, S. Tapaswi, S. Yousef and M. Cole, "The impact of mobility and node capacity on voice traffic," *International Journal of System Assurance Engineering and Management,* vol. 8, no. 33, pp. 1 - 9, March 2017.

[12] Ahmed Saihood, Al-Shaher, M. Fadhel, " A New Tiger Beetle Algorithm for Cybersecurity, Medical Image Segmentation and Other Global Problems Optimization," *Mesopotamian journal of Cybersecurity 4(1), pp. 17–46*, 2024.

[13] s. Cheng, P. Chen and C. Lin, "Traffic-Aware Patching for Cyber Security in Mobile IoT," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 29 - 35, 2017.

[14] W. Hu and Cao, "Quality-Aware Traffic Offloading in Wireless Networks," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 16, no. 11, March 2017.

[15] A. Msolli, A. Helali and H. Maaref, "Image encryption with the AES algorithm in wireless sensor network," Tunisia, July 2016.

[16] Bansod, G., Raval, N. & Pisharo, N., 2015. Implementation of a New Lightweight Encryption Design for Embedded Security. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* 10(1), pp. 142-151.

[17] Qashou, A., Yousef, S., Okoro, A. and Hazzaa, F. (2023), "Microgrid TestBed for Temporal Forecasting Patterns of Failure for Smart Cities", Technology and Talent Strategies for Sustainable Smart Cities, Emerald Publishing Limited, Leeds, pp. 189-227. https://doi.org/10.1108/978-1-83753-022-920231010

[18] Y. Zou, J. Zhu, X. Wang and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1727-1765, Sept. 2016.

[19] Emmanouis and Christos, "Security model for emergency real-time communications in autonomous networks," *Springer-Information Systems Frontiers, A Journal of Research and Innovation,* vol. 14, no. 3,pp.541–553[online]:https://link.springer.com/article/10.1007/s10796-010-9259-8, 2012.

[20] Alamsyah, A. Bejo and T. Adji, "AES S-box construction using different irreducible polynomial and constant 8-bit vector," Taiwan, 2017.

[21] M. Nagendra and M. C. Sekhar, "Performance Improvement of Advanced Encryption Algorithm using Parallel Computing," *International Journal of Software Engineering and Its Applications*, vol. 8, no. 2, pp. 287-296. [availble online]: https://pdfs.semanticscholar.org/. 2014.

[22] F. Hazzaa, S. Yousef, E. Sanchez and M. Cirstea, "Lightweight and Low-Energy Encryption Scheme for Voice over Wireless Devices," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp. 2992-2997.

[23] Sahu, S. K. & Kushwaha, A., 2014. Performance analysis of Symmetric Encryption algorithm for Mobile ad hoc networks. International Journal of Emerging Technology and Advanced Engineering, 4(6), pp. 619-624.

[24] A. Prakash, M. Satish, T. Sai and M. G., "Improving Cloud Security Using Multi Level Encryption and Authentication," *International Journal of Innovative Research in Information Security (IJIRIS),* vol. 2, no. 8, pp. 1-8, Aug. 2015.

[25] D. Salama and M. Hadhoud, "Evaluating the effect of symmetric Algorithms on Power consumption for Different Data types," *International Journal of Network Security*, vol. 11, no. 2, pp. 78-87, 2010.

[26] F. Hazzaa, S. Yousef, N. H. Ali and E. Sanchez, "The Effect of Nodes Density on Real Time Traffic in Mobile Ad Hoc Network," *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3),*pp. 209-212, London, United Kingdom, 2019.

[27] Stallings, W., Cryptography and Network Security: Principles and Practice (7th Edition). 7 th ed. Harlow- England: Pearson Education Limited, 2017.

[28] B. Mohd, T. Hayajneh, A. Vasilakos "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues" Journal of Network and Computer Applications, vol. 58, pp. 73-93, 2015.

[29] G. Bansod, N. Raval and N. Pisharo, "Implementation of a New Lightweight Encryption Design for Embedded Security," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 10, no. 1, pp. 142-151, 2015.

[30] Behnam Dezfouli, I. Amirtharaj, C. Li, “An energy measurement platform for wireless IoT devices," *Journal of Network and Computer Applications*, vol. 121, pp. 135-148, 2018.

[31] F. Hazzaa and S. Yousef, "Performance Analysis for Traffics in Mobile Ad Hoc Network," in *11th International Conference on Global Security, Safety & Sustainability ICGS3 - Springer International Publishing AG*, London, 2017.

[32] A. Popov, "Prohibiting RC4 Cipher," Internet Engineering Task Force, WA, USA, 2015.

[33] P. Gope, T. Hwang, " A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application in WSN," IEEE Transactions on Industrial Electronics, Industrial Electronics Society, vol. 63, no. 11, pp. 7124-7132, 2016.

[34] Z. A. Abduljabbar, A. Ibrahim, M. A. Al Sibahee, S. Lu and S. M. Umran, "Lightweight Privacy-Preserving Similar Documents Retrieval over Encrypted Data," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2021, pp. 1397-1398, doi: 10.1109/COMPSAC51774.2021.00202.