JJEE
Jordan Journal of Electrical Engineering
Volume 8, Number 4, December 2022
Pages 339-354
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

Design and Implementation of a Lightweight and Fast Tiny Advanced Encryption Standard Algorithm

Alaa S. Abdalrazzaq[1], Salah Abdulghani Alabady[2*]

[1, 2] Department of Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq
E-mail: eng.salah@uomosul.edu.iq

Received: July 24, 2022 *Revised: September 16, 2022* *Accepted: September 24, 2022*

***Abstract*—** Cryptographic algorithms are gaining importance due to their relevance and importance in the areas of privacy and security experienced by the Internet of Things (IoT) devices. They improve data privacy and confidentiality by limiting who can decrypt the data to the person who has the key. Advanced encryption standard (AES) is one of the most important encryption algorithms in use. This algorithm uses 10 rounds for each encryption and decryption process and encrypts data starting with 16 bytes, which increases the time needed for encryption and decryption. In order to speed up encryption and decryption while maintaining security levels more than or equal to those offered by the AES algorithm, this paper proposes an algorithm called tiny advanced encryption standard (TAES) with two different scenarios. As the encryption process starts with 4 bytes and goes up to an unlimited number of bytes, TAES features are less complex and more flexible than those of AES and can be quickly implemented, and are characterized by high encryption of images, texts, and sounds. The proposed TAES algorithm is tested on text and images using the MATLAB software. The result is completely distorted images and text. The test results also unveil that the encryption and decryption speeds as well as the throughput of the proposed TAES is much better than those of the original AES algorithm. Moreover, a low signal-to-noise ratio (SNR) values are obtained, indicating a greater degree of image distortion as a consequence of utilizing the proposed TAES algorithm.

***Keywords*—** Advanced encryption standard algorithm; Internet of things; Throughput; Privacy and security of IoT; Cryptography techniques; Signal-to- noise ratio.

1. INTRODUCTION

Kevin Ashton of Procter and Gamble is the creator of the term "Internet of Things" (IoT) [1]. Since then, thanks to the growth of the IoT, connectivity of smart devices has improved. Thermostats, smart watches, cellphones and house lights are some examples of these smart gadgets. The network and the sensor components are the two components that make up the IoT system. It is possible to evaluate and keep track of environmental conditions using the sensor component. Data is sent and exchanged across the network between the server and the IoT component via the network component [2]. Some IoT-related businesses continue to market and sell insecure and cumbersome IoT devices. Recent IoT device protocols lack key security features, and there is hardly any device trust [3].

One of the most important methods for preventing cyberattacks on transferred data is encryption algorithms. Encryption is a method for keeping the original text of a message hidden so that only the intended users may read it. Therefore, encryption is a security technology to guard sensitive data and information sent over wireless networks [4]. The classification of the encryption methods that are employed is also affected by the kind of key, so encryption methods are divided into symmetric encryption and asymmetric encryption [5-7]:

* Corresponding author
Article's DOI: 10.5455/jjee.204-1658696772

- **Symmetric encryption**: symmetric encryption uses a single key for encryption and decryption. Rivest Cipher 4 (RC4), data encryption standard (DES), 3DES, RC5, Blowfish, and advanced encryption standard (AES) are examples of this type of encryption, the model of which is shown in Fig. 1.

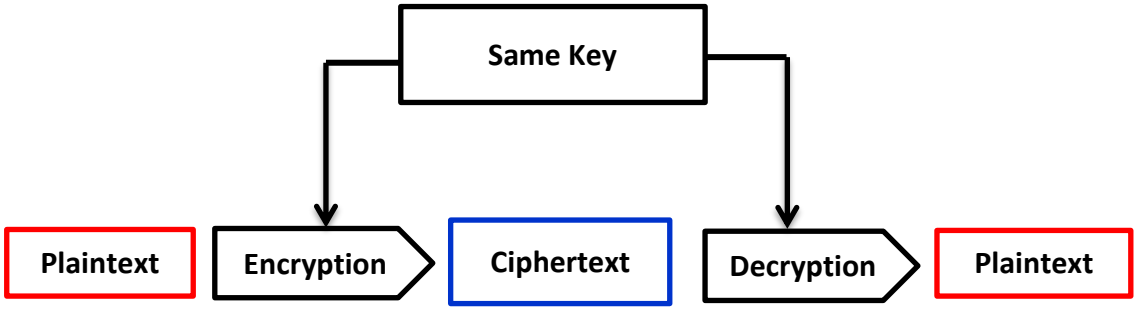

Fig. 1. Symmetric encryption model.

- **Asymmetric encryption:** asymmetric encryption uses a public key for encryption and a private key for decryption. Elliptic curve cryptosystem (ECC), Rivest-Shamir-Adelman (RSA), and digital signatures are examples of this type of encryption, the model of which is shown in in Fig. 2.

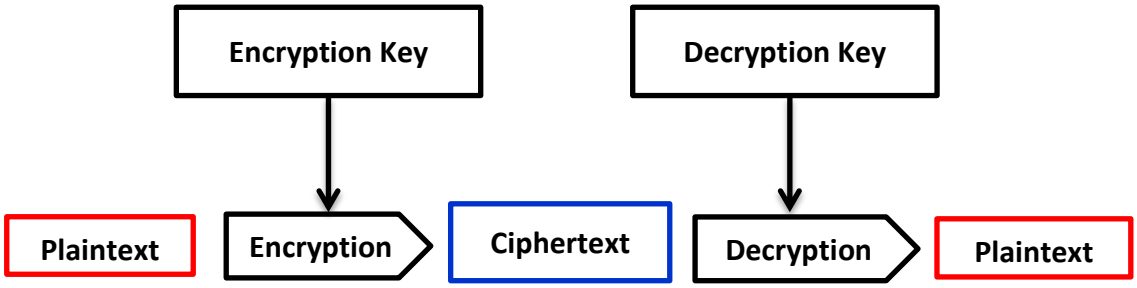

Fig. 2. Asymmetric encryption model.

One of the symmetric encryption algorithms that use the same key for both encryption and decryption is the advanced encryption standard (AES) algorithm. The entries range in size from 16 to 24 to 32 bytes. As a result, using this algorithm for small data sizes is not flexible. As a result, an enhancement to this algorithm is proposed in this paper. Here, the number of rounds has been decreased to 6 rounds with small data sizes starting at 4 bytes to speed up encryption and decryption while maintaining security levels more than or equal to those offered by the original AES algorithm.

The rest of this paper is organized as following: section 2 reviews recent research on AES and encryption algorithms. Section 3 explains the AES algorithm and its features and provides more information on the proposed algorithm and related scenarios. Section 4 includes the results of the proposed algorithm. Finally, section 5 includes the conclusion and future works.

2. RELATED WORK

Because there are Internet-connected gadgets all around us that sense user data and transmit it from one place to another, privacy has become crucial for protecting sensitive data sent over the Internet from theft. The authors in [8], explained the definition of privacy

and various privacy violations are explained, along with the various layers of IoT security architecture that are available and how they work to safeguard the availability, confidentiality and security of IoT devices.

The IoT layers, the risks to each layer, and issues related to the sensing layer, network layer, middleware layer, gateways and application layer are explained in [9], which also addressed potential solutions for IoT. Advanced computing, fog computing, blockchain, and the latest IoT security and machine intelligence technologies are discussed as well.

To determine the best algorithm for usage in the future, the work presented in [6] includes performance evaluations of symmetric and asymmetric cryptographic algorithms. By using several file formats, including binary, document and images, it evaluates both symmetric encryption techniques (AES, DES, and Blowfish) and an asymmetric encryption algorithm (RSA). Evaluation criteria such as throughput, encryption time, and decryption time were used to compare these different encryption algorithms. The obtained simulation results showed that AES outperforms other algorithms in terms of throughput and decryption speed.

The implementation of AES encryption and decryption - using a key length of 128 bits and MATLAB program - to offer a secure communication channel for data transmission of images and texts was presented in [10]. The output was shown to be encrypted in a way that prevents message detection by unauthorized users. There is no leniency in distortion, and the decrypted output is identical to the input. This shows the excellent efficiency of the AES algorithm because it prevents data loss during transmission while allowing the recovery of the original message as an output for decryption

Replacing the AES MixColumns transformation with the bit-shifting to increase the efficiency of AES was proposed in [11]. Text files and images were encrypted to gauge how well the modified AES algorithm performed in terms of encryption time, CPU consumption, and avalanche impact. According to the achieved results, the enhanced AES - which has a quicker encryption time for files and text images - improves the effectiveness of the normal AES. Compared to the standard AES, the modified AES used less CPU. The new method also resulted in a greater avalanche effect during testing

A technique for effectively combining the AES encrypted and decrypted AES into a fully operational AES crypto-engine is proposed in [12]. The general architecture is simple and suitable for smart card applications or other hardware-dependent ones. Hardware complexity has decreased to 57% of its original level.

The AES algorithm is characterized by the complexity of the encryption and decryption process, as it takes a long time to execute. In [13], the AES algorithm was modified and the column-mix step was eliminated, which reduced the complexity of the algorithm and speeded up the execution time.

Using a 200-bit text and a 400-bit encryption key, the AES-AES algorithm was proposed by the researchers in [14] by eliminating the column-mix step and splitting the key into two 200-bit keys, which reduced the time complexity of execution.

In in [15], the security of IoT systems' data transfer has improved. The AES algorithm with a dynamic key generation approach with a frame containing 16 bytes of data that are delivered sequentially (it will lead to significant changes if followed in this order) is

proposed. This technique uses encryption technology to prevent data theft and is used when sending data over the IoT online shopping, and stock trading.

The entire history of the AES algorithm is described in [16], along with details of how it works. The paper revealed that AES was developed in cooperation between the National Institute of Standards and Technology (NIST) and the global cryptography community with the ultimate objective of developing the Federal Information Processing Standard (FIPS), and that AES technology is expected to be used by the US government and businesses, according to NIST.

The majority of earlier research focused on enhancing the AES algorithm's performance. This paper will present a new method to enhance the performance of the original AES algorithm by making it more flexible for small data, where the input starts at 4 bytes, and by simplifying its implementation by removing some steps, which reduces the complexity and power consumption of IoT devices. This is achieved while maintaining a security level greater than or equivalent to the original AES algorithm's security level.

3. THE PROPOSED METHOD

Before going into the details of the proposed algorithm, the AES algorithm and its steps will be explained in this section because it forms the foundation of the proposed algorithm.

3.1. Advanced Encryption Standard

The AES method - discovered by the Belgian cryptographic experts, Vincent Regmen and Joan Daemen [17] - was chosen as the strongest encryption technology based on previous research and analysis. It is a changeable block cipher that employs a symmetric key with a changeable key length of 128, 192, or 256 bits [6]. Depending on the size of the encryption key, the algorithm encrypts data blocks of 128 bits in 10, 12, and 14 rounds [18]. The characteristics of this algorithm are the following [19, 20]:

- Execution speed and code compression
- Simplicity of design
- Resistance and protection against known attacks
- Efficiency in software and hardware
- Usage - particularly in small devices - across many platforms.

The AES algorithm operates in four steps [6, 19]:

- Substituting Byte (sub byte) step
- Shifting Rows step
- Mixing columns step
- Adding Round Key step

3.2. The Proposed Method

The main objective of this paper is to propose a lightweight and fast encryption algorithm based on the AES algorithm called tiny AES (TAES) algorithm. As opposed to the original AES, this proposed algorithm tries to implement text and image encryption rapidly.

It reduces the power consumption used by heterogeneous internet devices, which have processor, memory, and power limitations [3]. The TAES is proposed based on 4 bytes data size as input data for the algorithm and the key size is only 4 bytes and the number of rounds is only six. The encryption key has a 4-byte length. Similar to the key expansion steps in the original AES algorithm, the key is extended in steps. The proposed algorithm's key expansion step is shown in Fig. 3(a) and the (g) sub-function used in the expansion is shown in Fig. 3(b).

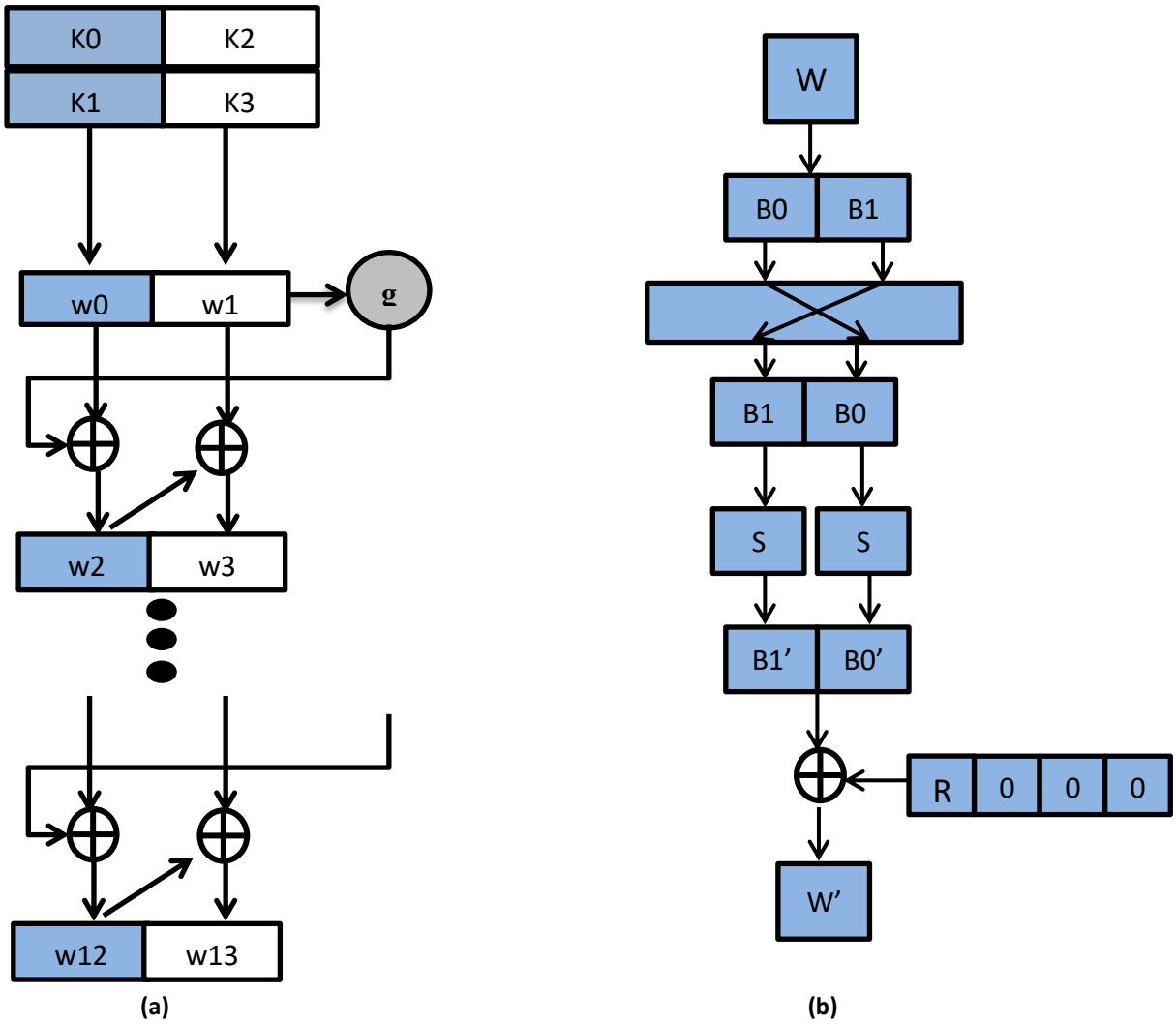

Fig. 3. TAES key expansion model: a) key expansion function; b) sub-function of (g).

3.2.1. TAES First Scenario

The first scenario includes removing the Mix Column step in the AES from all rounds and adding a new step to the algorithm, making it more difficult for hackers to access or alter data by making the encryption process more complex. Round 0 is often referred to as the initial round. The first four bytes (w0 and w1) of the key are used to perform a simple XOR operation on the input array. The steps are given in the following order for the first five rounds:

- **Substitute Bytes**: Referred to as sub bytes. It is a permutation stage in which the entries of a predetermined (16 × 16) matrix called substitution box (S-box) are substituted for the contents of each cell of the input array as shown in Fig. 4.

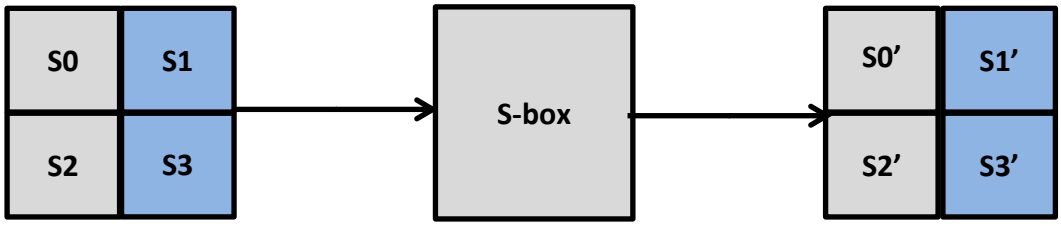

Fig. 4. TAES sub-byte step.

- **Shift Row:** first row of the state matrix remains unchanged. A 1-byte circular left shift is performed for the second row as shown in Fig. 5.

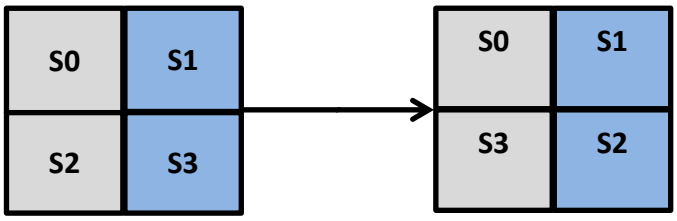

Fig. 5. TAES shift row step.

- **Add Round Key**: In the state matrix, 32 bits are bitwise XORed with a portion of the expanded key as shown in Fig. 6.

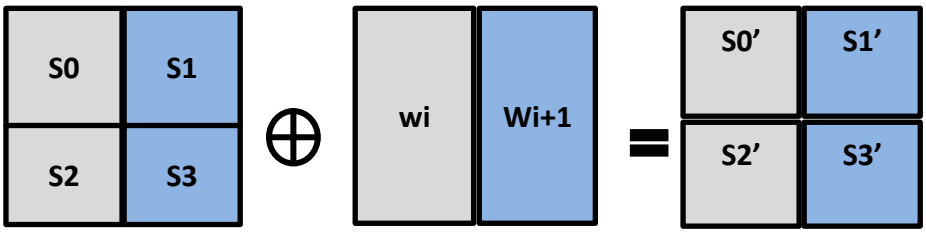

Fig. 6. TAES Add Round Key step.

- **XORing Step:** In the sixth round, this new step is added to the algorithm steps. This step contains a simple XORing operation on the matrix produced by the Add Round Key step with the XORing matrix. This matrix is derived from the original matrix used in the original algorithm's mix column step, as shown in Fig. 7.

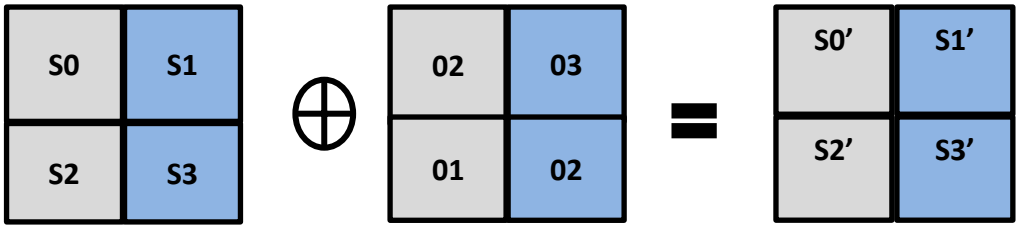

Fig. 7. TAES XORing step.

The decryption process starts with a simple XORing step that includes the ciphertext received from the encryption process with the XORing matrix used in the XORing step of the sixth round of encryption. Then, the initial Add Round Key step begins, followed by the six rounds, each round consisting of the following steps:

- **Inverse Shift Row:** Also referred to as (Inv Shift Rows), involves shifting circularly in the opposite direction while shifting the second row's first byte to the right by one.
- **Inverse Substitute Byte**: Also referred to as (Inv Sub Byte) a new byte is allocated from the inverted s-box table for each byte of the matrix.
- **Add Round Key:** The round adds key's inverse transformation is the same as the round key's forward transformation because the XOR operation is its inverse.

3.2.2. TAES Second Scenario

This scenario is similar to the original AES algorithm in terms of constructing the rounds except for replacing the mix column step with the XORing step with a matrix similar to the matrix used in the first scenario. Before starting with the six rounds there is an initial round, which is a simple XOR operation between the input array and the input key.

The encryption and decryption processes for the TAES first and second scenarios are shown in Figs. 8 and 9, respectively, and Fig. 10 shows the pseudo-code of the encrypting process for the two proposed algorithm scenarios.

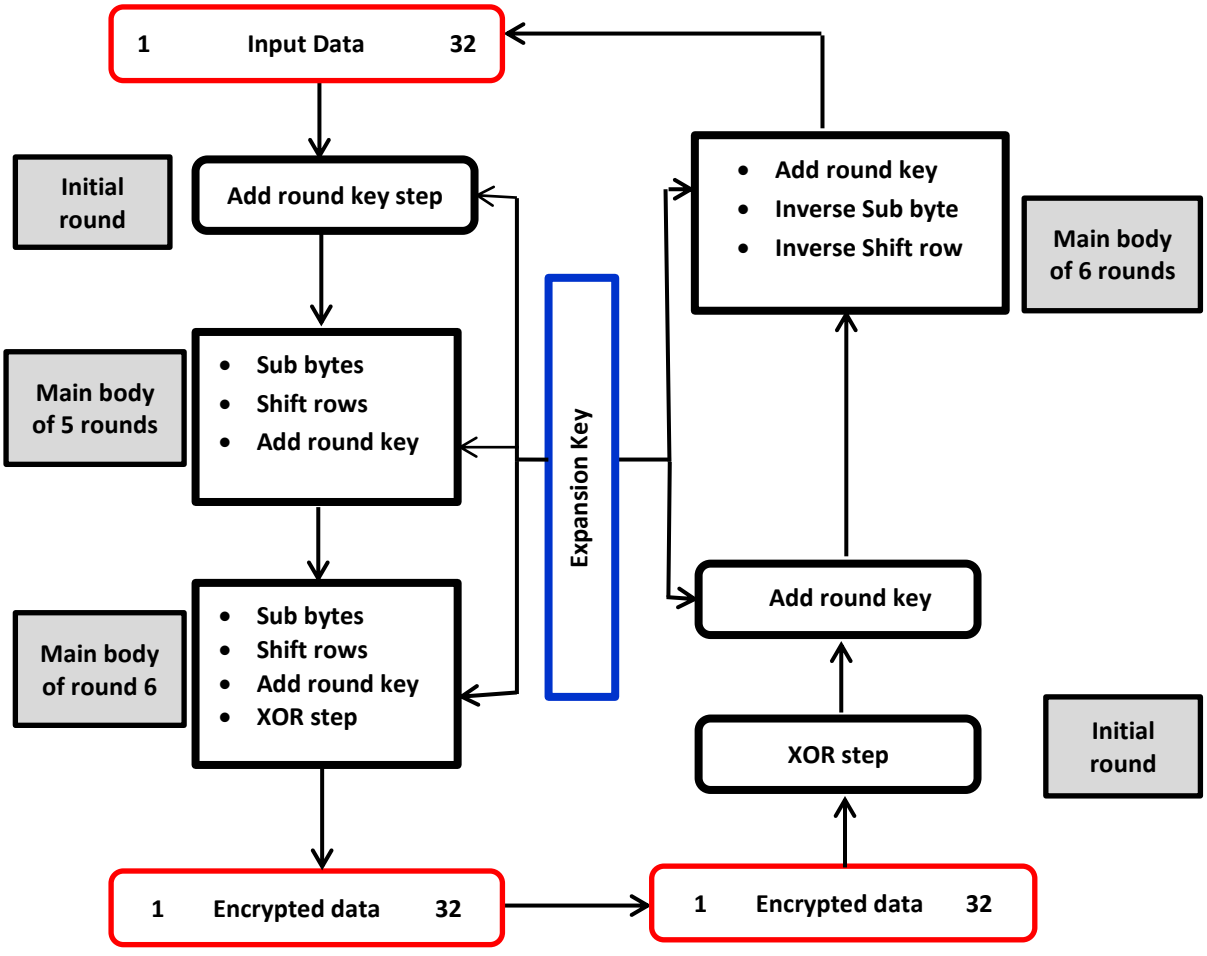

Fig. 8. TAES first scenario.

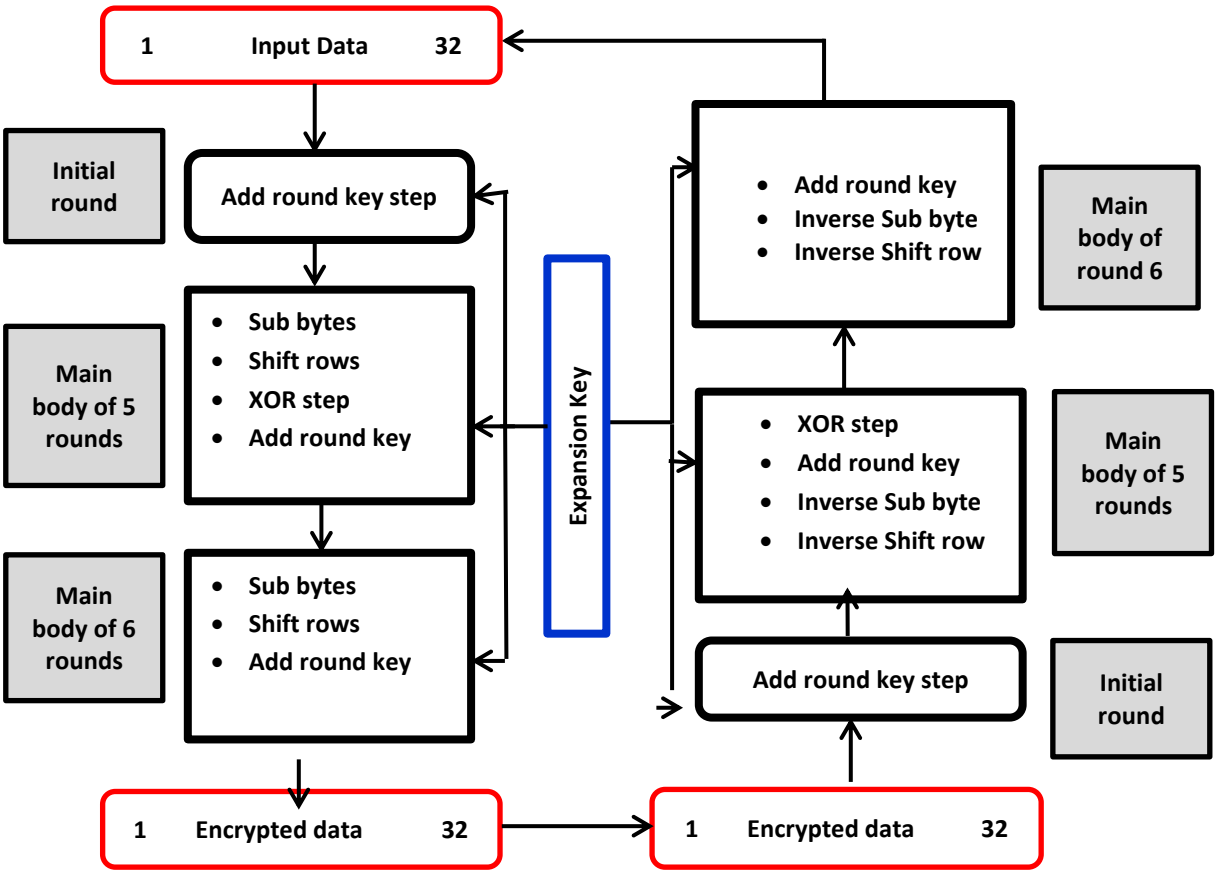

Fig. 9. TAES second scenario.

4. SIMULATION RESULTS ANALYSIS

The performance of the TAES was evaluated by two types of inputs; texts and images. The MATLAB program is used for the implementation and evaluation of the TAES. The performance of the TAES is evaluated according to the following criteria:

- **Encryption Time**: The length of time needed to transform plaintext into ciphertext. The encryption time is measured in seconds and depends on the size of the message block and the key. It directly affects how well the encryption algorithm performs. Every cryptographic algorithm had a minimum encryption time required to create a quick and responsive encryption system.
- **Decryption Time**: The amount of time needed to decipher ciphertext and find the plaintext. The decryption time should be less similar to the encryption time and measured in seconds to make the cryptographic algorithm quick and responsive.
- **Throughput:** It is the proportion of messages that are successfully delivered across a communication channel in telecommunications networks. The throughput rate, which is the speed of the communication channel, is typically expressed in kilobytes (KB) per second. By dividing the total size of the encrypted block (Tp) by the total encrypting time (Et), one can determine the transmission rate of an encryption technique. The power consumption of the algorithm will decrease as throughput grows. Eq. (1) represents the throughput.

$$\text{Throughput} = \frac{Tp}{Et} \quad (1)$$

- **Visual Distortion**: The process of trying to interpret the original text from ciphertext or to perceive the specifics of the original image from a distorted image by looking.
- **Signal-to-Noise Ratio (SNR):** This is a measurement that contrasts the strength of the desired signal with the strength of noise in science and engineering. SNR is frequently stated in decibels (dB) as a metric for describing an image's quality. Eq. (2) represents the SNR.

$$SNR = 10\,log_{10}\frac{P_{signal}}{P_{noise}} \quad (2)$$

4.1. The First Test

Text testing includes entering texts in the following byte sizes: 4, 8, 16, 64, 128, 256, 1024, and 2040. There are letters, numbers, symbols, and spaces in these texts. The encryption and decryption times as well as the throughput are then calculated for each scenario and compared to the original AES algorithm. The first and second scenarios' encryption and decryption timings and throughput are displayed in Tables 1 and 2, respectively, along with a comparison to the original AES algorithm.

Table 1. Encryption and decryption times and throughput of both the TAES first scenario and the original AES.

Data Size [byte]	Encryption Time of TAES 1st scenario [s]	Decryption Time of TAES 1st scenario [s]	Encryption Time of AES [s]	Decryption Time of AES [s]	Throughput of TAES 1st scenario [bit/s]	Throughput of AES [KB/s]
4	0.002733	0.001018	-----	-----	1.44	-----
8	0.002727	0.001635	-----	-----	2.86	-----
16	0.002153	0.000822	0.019834	0.021726	7.26	0.788
64	0.001900	0.000789	0.011651	0.014639	32.89	5.36
128	0.001915	0.000741	0.011868	0.013324	65.27	10.532
256	0.001894	0.000740	0.012943	0.014431	131.99	19.315
1024	0.002057	0.000770	0.013145	0.013305	486.14	76.07
2040	0.002025	0.000955	0.011101	0.014318	983.79	179.46

Table 2. Encryption and decryption times and throughput of both the TAES second scenario and the original AES.

Data Size [byte]	Encryption Time of TAES 2d scenario [s]	Decryption Time of TAES 2d scenario [s]	Encryption Time of AES [s]	Decryption Time of AES [s]	Throughput of TAES 2d scenario [bit/s]	Throughput of AES [KB/s]
4	0.000534	0.000466	-----	-----	7.32	-----
8	0.000392	0.000389	-----	-----	19.93	-----
16	0.001853	0.000670	0.019834	0.021726	8.43	0.788
64	0.001698	0.000679	0.011651	0.014639	36.81	5.36
128	0.001659	0.000669	0.011868	0.013324	75.35	10.532
256	0.001657	0.000669	0.012943	0.014431	150.88	19.315
1024	0.002138	0.000870	0.013145	0.013305	468.38	76.07
2040	0.001960	0.000777	0.011101	0.014318	1016.42	179.46

The algorithm's input: Plaintext includes text with a size range from 4 bytes to 2040 bytes, or images, or audio (after converting them to matrix).

Begin

"Initialization steps"

1. Insert plaintext
2. Insert encryption key "32 bit"
3. The proposed algorithm contains four scenarios

"Initial steps"

4. Reshape an input matrix (plaintext) to a component matrix 2×2.
5. Reshape encryption key to matrix 2×2.
6. Execution step AddRoundKey between plaintext matrix and key matrix
7. AddRoundKey is a simple xor operation between the input bits and the key bits
8. The output of the last step is a 2 × 2 matrix ready to be entered into the six steps

"proposed algorithm rounds"

"encryption process"

9. If (proposed algorithm scenario = first scenario)
   {
10. For(i=1;i<5;i++)
    {
11. Replace the bytes of the resulting array from the AddRoundKey step with bytes from the S-BOX
12. Matrix's first row that remain unchanged. A 1-byte circular left shift is performed for the second row
13. The matrix's 32 bits are bitwise XORed with a portion of the expanded key.
    }
14. End for

"Step of the final round"

15. Replace the bytes of the resulting array from the AddRoundKey step with bytes from the S-BOX
16. Matrix's first row that remains unchanged. A 1-byte circular left shift is performed for the second row
17. The matrix's 32 bits are bitwise XORed with a portion of the expanded key.
18. simple XOR operation on the matrix produced by the previous step with the XORing matrix
    }
19. End if
20. If(proposed algorithm scenario = second scenario)
    {
21. For(i=0;i<5;i++)
    {
22. Replace the bytes of the resulting array from the AddRoundKey step with bytes from the S-BOX
23. Matrix's first row that remain unchanged. A 1-byte circular left shift is performed for the second-row
24. simple XORing operation on the matrix produced by the previous step with the XORing matrix
25. The matrix's 32 bits are bitwise XORed with a portion of the expanded key.
    }
26. End for

"the step of the final round"

27. Replace the bytes of the resulting array from the AddRoundKey step with bytes from the S-BOX
28. Matrix's first row that remain unchanged. A 1-byte circular left shift is performed for the second row
29. The matrix's 32 bits are bitwise XORed with a portion of the expanded key
    }
30. End if

The algorithm's output: Unreadable encrypted text, distorted images in which the features are hidden or unclear audio.

Fig. 10. The pseudo-code of the encrypting process for the two proposed algorithm scenarios.

Figs. 11 and 12 compare encryption and decryption times of the proposed algorithm's first and second scenarios to those of the original AES algorithm.

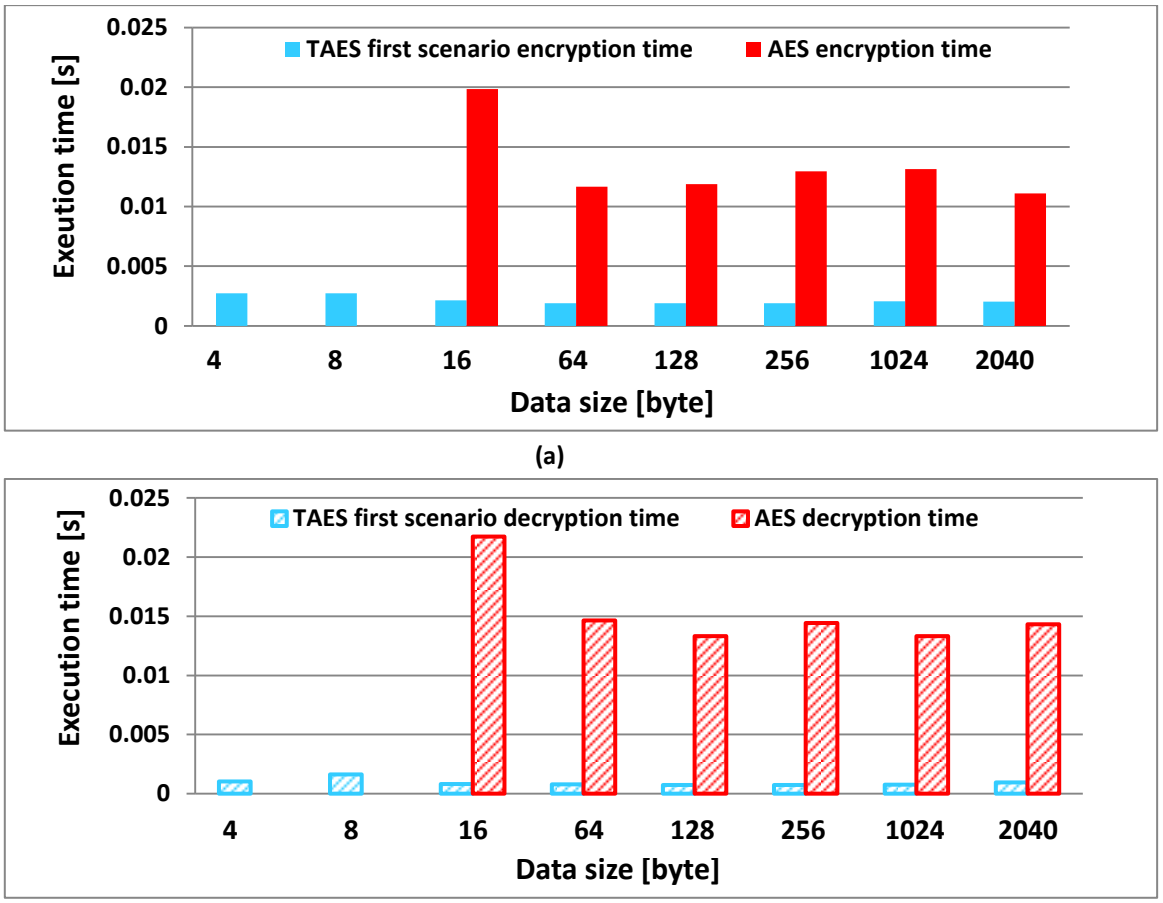

Fig. 11. a) Encryption time; b) decryption time of TAES first scenario and the original AES.

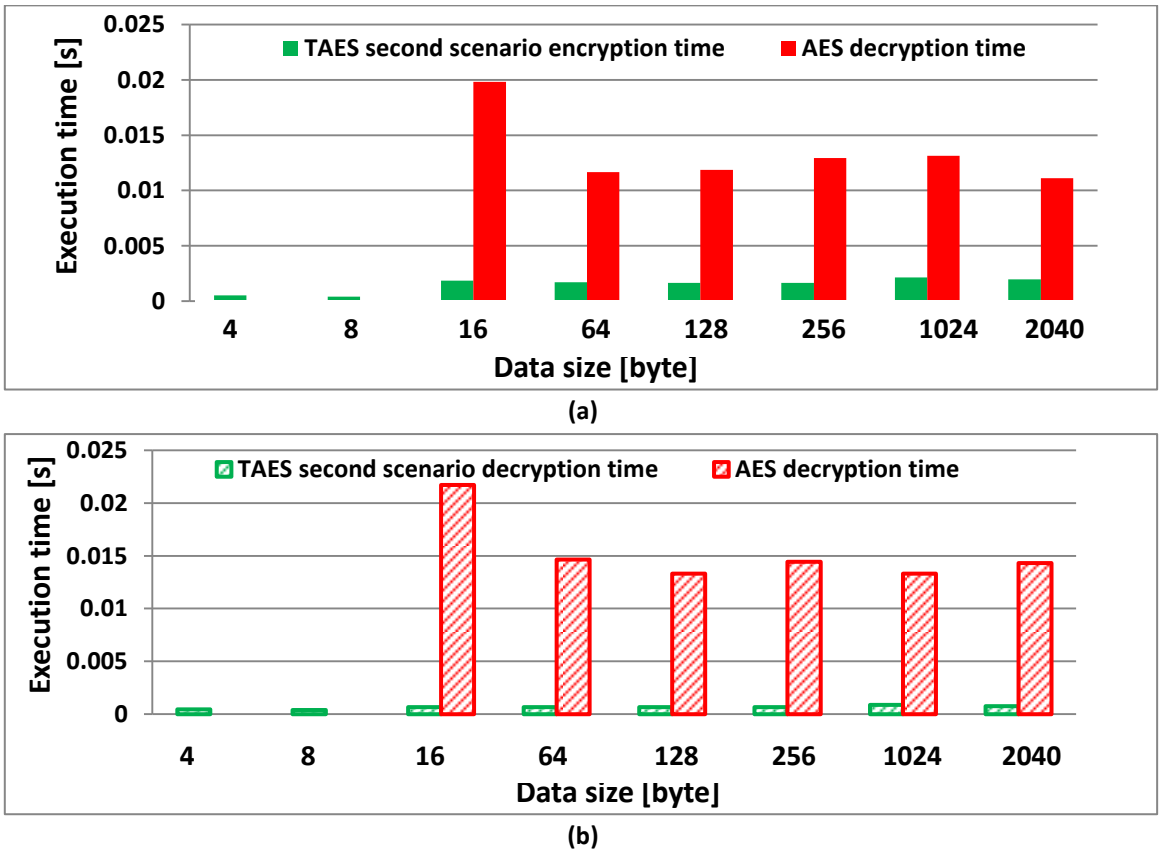

Fig. 12. a) Encryption time; b) decryption time of TAES second scenario and the original AES.

The encryption and decryption time of the proposed algorithm is quicker than the original AES algorithm. This is due to the proposed TAES algorithm's decision to minimize the number of rounds, which increased the speed at which encryption and decryption were implemented. Fig. 13 compares the throughput of the proposed algorithm with the original AES algorithm.

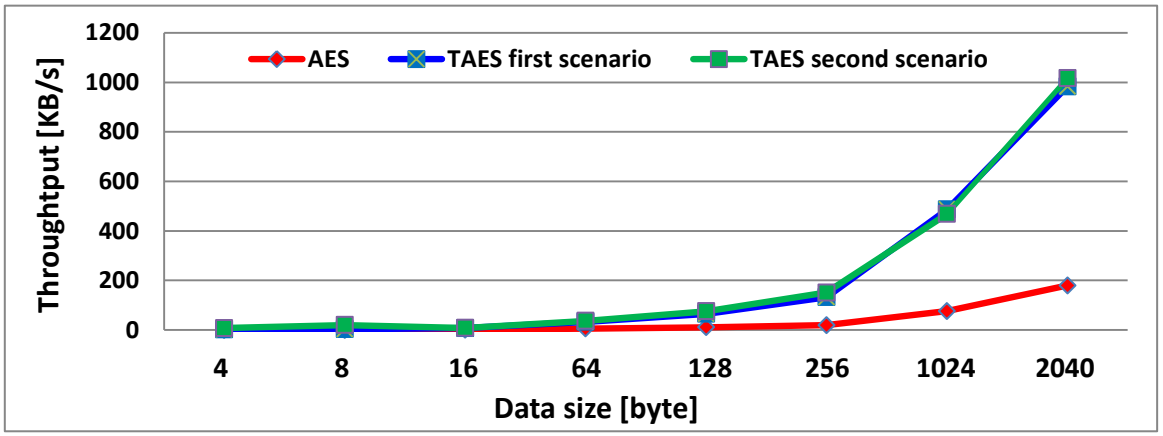

Fig. 13. Throughput of the proposed TAES scenarios and the original AES.

The results in Fig. 13 show that the proposed algorithm is superior to the original AES algorithm in terms of throughput, as the amount of data sent using the first and second scenarios is higher compared to the same amount of data sent using the original AES algorithm. Fig. 14 displays the character message output in plain text, encrypted form and decrypted form. To carry out the encryption and decryption procedure, each character is mapped to the proper integers (converted character to ASCII values) and then transformed into a (2 × 2) matrix.

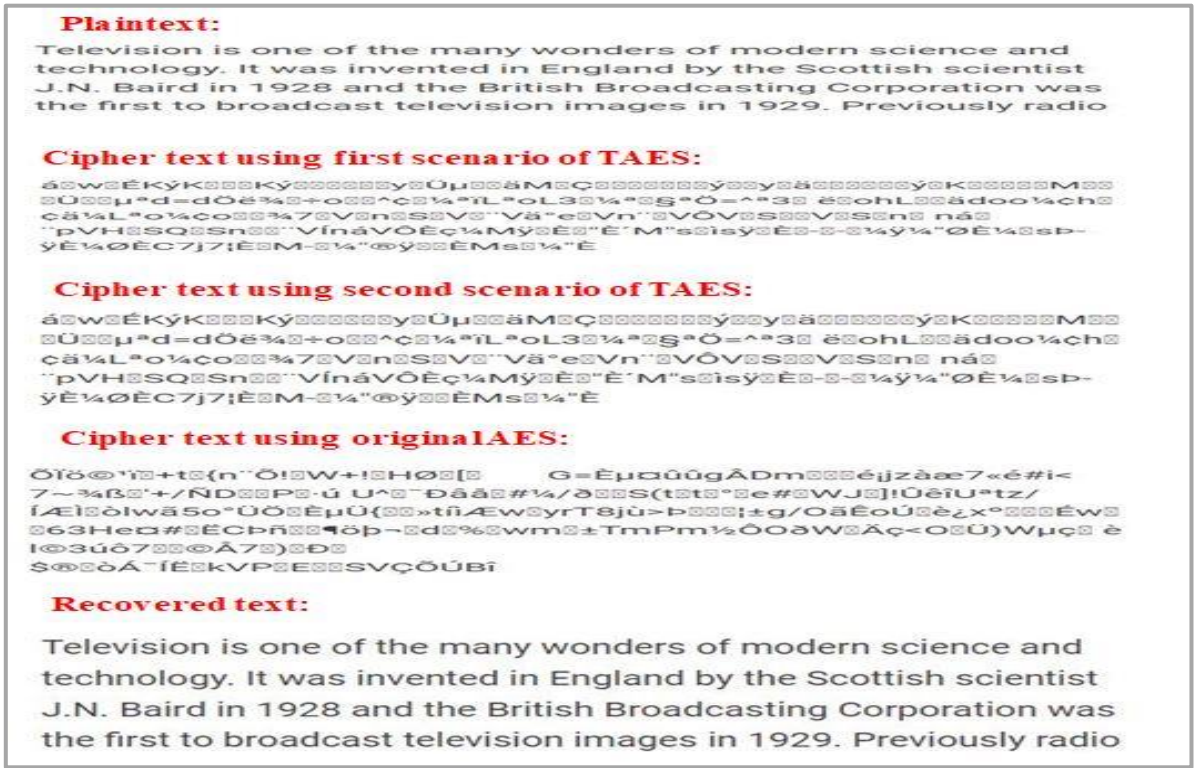

Fig. 14. Results of text encryption and decryption using the proposed TAES algorithm.

4.2. The Second Test

Inserting images of different sizes is a part of the image testing process. Then, the speed of the encryption and decryption processes is computed, as well as the SNR, throughput and the human vision-to-picture distortion ratio. (2 × 2) matrices are taken for each of these images and fed into the encryption process, and then assembled to get the encrypted image. The decryption process is a reverse process where a (2 × 2) matrix of the image is taken and entered into the decryption process, and then the sub-arrays are grouped to get the recovered image. The purpose of image diversity is to test the efficiency of the proposed algorithm for images of different types and sizes.

Fig. 15 displays a black-and-white image of Barbara with a size of (160 × 160) pixels. It demonstrates how the first and second scenarios' encryption produces a completely distorted image from which no details could be recognized. The resulting encrypted image is also colorful. In a second comparison, Barbara's uncolored image with dimensions of (160 × 160) pixels (12 KB) is used to test the effectiveness of the proposed algorithm. Table 3 lists the SNR values, as well as the encryption time, decryption times and throughput for the TAES scenarios and the original AES algorithm for the Barbara image.

Table 3 shows that the proposed algorithm's encryption and decryption times are quicker than those of the original AES algorithm. This is because the proposed algorithm is simpler than the original AES algorithm. Although the proposed algorithm's throughput value is larger than that of the original AES algorithm, the SNR ratio is comparable to the algorithm's values, which is a primary evidence for the image's high rate of distortion.

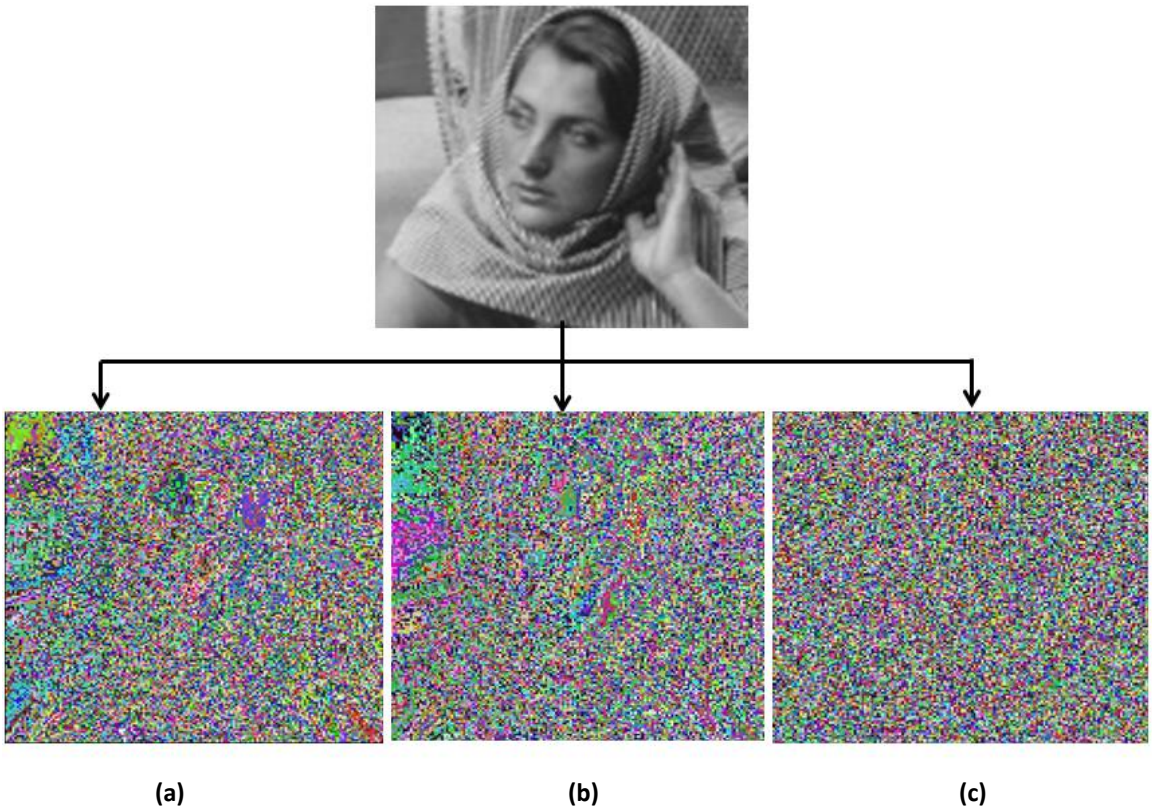

Fig. 15. Encrypted Barbara image using the original AES and TAES scenarios: a) TAES first scenario; b) TAES second scenario; c) original AES.

Table 3. SNR, encryption time, decryption time and throughput for Barbara image encrypted using the TAES scenarios and original AES algorithm.

Items	SNR [dB]	Encryption time [s]	Decryption time [s]	Throughput [KB/s]
TAES 1st scenario	3.4002	0.001439	0.001418	8339.12
TAES 2nd scenario	3.5411	0.001545	0.002115	7766.99
AES	3.5989	0.011749	0.017287	1021.36

In a second comparison, the parrot's color image – shown on Fig. 16 - with dimensions of (232 × 212) pixels (27.2 KB) is used to test the effectiveness of the proposed algorithm. It demonstrates how the first and second scenarios' encryption produces a completely distorted image from which no details could be recognized. The resulting encrypted image is also colorful.

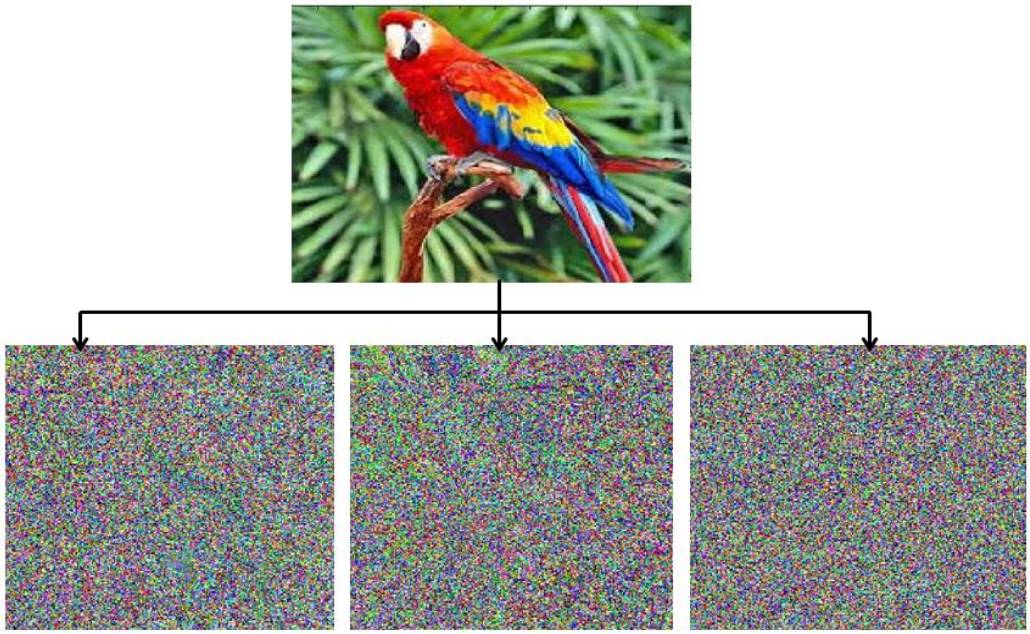

Fig. 16. Encrypted parrot image using the original AES and TAES scenarios: a) TAES first scenario; b) TAES second scenario; c) original AES.

Table 4 lists the SNR values, as well as the encryption time, decryption time and throughput for the TAES scenarios and the original AES algorithm for the parrot image.

Table 4. SNR, encryption time, decryption time, and throughput for the parrot image encrypted using the TAES scenarios and original AES algorithm.

Items	SNR [dB]	Encryption time [s]	Decryption time [s]	Throughput [KB/s]
TAES 1st scenario	1.735	0.002109	0.001935	12897.10
TAES 2nd scenario	1.6666	0.002329	0.001966	11678.83
AES	1.7259	0.011432	0.018546	2379.28

Table 4 shows that the encryption and decryption times of the proposed algorithm are faster than those of the original AES algorithm. This is because the proposed algorithm is

simpler than the original AES algorithm. Although the throughput value of the proposed algorithm is greater than that of the original AES algorithm, the SNR ratio is close to its value in the original AES algorithm, which constitutes a basic evidence of a high rate of image distortion.

5. CONCLUSIONS

The AES algorithm is one of the most effective encryption algorithms used in encryption systems to ensure information security during transmission. However, putting this method into practice is difficult and takes longer than anticipated. Therefore, this paper proposed the TAES algorithm as an improvement to the original AES algorithm. The proposed algorithm provided flexibility in data encryption, allowing for encryption of data ranging in size from 4 bytes to an infinite number of bytes. A tiny encryption key with only 4 bytes and 6 rounds is also present. The obtained results proved the algorithm's flexibility for use with both small and large data sets, and its capability of producing fully encrypted images and texts that could be recovered without losing any of their details. Additionally, the proposed algorithm's encryption and decryption times were faster than those of the original AES algorithm, which ultimately increased throughput or the number of bits transmitted per second. Additionally, the SNR value has decreased from that of the original AES algorithm, indicating a greater degree of image distortion when employing the proposed algorithm. As a future work, more scenarios will be proposed to simplify the algorithm, shorten encryption and decryption times, and improve security and output.

REFERENCES

[1] I. Butun, P. Osterberg, H. Song, "Security of the internet of things: vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[2] M. Jian, Y. Cheng, C. Shen, "Internet of things (IOT) cybersecurity based on the hybrid cryptosystem," *International Conference on Advanced Communication Technology*, pp. 176–181, 2019.

[3] T. Abdullah, W. Ali, S. Malebary, A. Ahmed, "A review of cyber security challenges, attacks and solutions for internet of things based smart home," *International Journal of Computer Science and Network Security*, vol. 19, no. 9, pp. 139–146, 2019.

[4] F. Mallouli, A. Hellal, N. Sharief Saeed, F. Abdulraheem Alzahrani, "A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms," *Proceedings of 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019,* pp. 173–176, 2019.

[5] M. Habib, M. Ahmad, S. Jabbar, S. Ahmed, J. Rodrigues, "Speeding up the internet of things: LEAIoT: a lightweight encryption algorithm toward low-latency communication for the internet of things," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6. pp. 31–37, 2018.

[6] M. Panda, "Performance analysis of encryption algorithms for security," *International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings*, pp. 278–284, 2017.

[7] A. Mandal, C. Parakash, A. Tiwari, "Performance evaluation of cryptographic algorithms: Des and AES," *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science: Innovation for Humanity*, pp. 1-5, 2012.

[8] H. Ali, W. El-Medany, "IoT security: a review of cybersecurity architecture and layers," *IET Conference Publications*, vol. 2019, no. CP758, pp. 4–10, 2019.

[9] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[10] V. Aparna, A. Rajan, I. Jairaj, B. Nandita, P. Madhusoodanan, A. Remya, "Implementation of AES algorithm on text and image using MATLAB," *2019 3rd International Conference on Trends in Electronics and Informatics*, pp. 1279-1283, 2019.

[11] H. Gamido, A. Sison, R. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, pp. 942–948, 2018.

[12] C. Lu, S. Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pp. 277–285, 2002.

[13] S. Arman, T. Rehnuma, M. Rahman, "Design and implementation of a modified AES cryptography with fast key generation technique," *IEEE International Women in Engineering Conference on Electrical and Computer Engineering*, pp. 191-195, 2020.

[14] A. Gupta, 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*, pp. 422–427, 2017.

[15] T. Dang, "Advanced AES algorithm using dynamic key in the internet of things system thanh," in *2019 IEEE 4th International Conference on Computer and Communication Systems*, pp. 682–686, 2019.

[16] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback, "Report on the development of the Advanced Encryption Standard (AES)," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, no. 3, pp. 511–577, 2001.

[17] S. Mukta, S. Azad, "Secure hash algorithm advanced encryption standard," *Practical Cryptography: Algorithms and Implementations Using C++*, vol. 6, no. 2. pp. 207–223, 2014.

[18] M. Anand Kumar, S. Karthikeyan, "Investigating the efficiency of blowfish and rejindael (AES) algorithms," *International Journal of Computer Network and Information Security*, vol. 4, no. 2, pp. 22–28, 2012.

[19] H. Gamido, A. Sison, R. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 942–948, 2018.

[20] M. Mushtaq, S. Jamel, A. Disina, Z. Pindar, N. Shakir, M. Deris, "A survey on the cryptographic encryption algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11 , pp. 333-344, 2017.