Hindawi
Security and Communication Networks
Volume 2023, Article ID 9892173, 1 page
https://doi.org/10.1155/2023/9892173

Retraction

Retracted: A Secure Environment Using a New Lightweight AES Encryption Algorithm for E-Commerce Websites

Security and Communication Networks

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] F. T. Abdul Hussien, A. M. S. Rahma, and H. B. Abdul Wahab, "A Secure Environment Using a New Lightweight AES Encryption Algorithm for E-Commerce Websites," *Security and Communication Networks*, vol. 2021, Article ID 9961172, 15 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9961172, 15 pages
https://doi.org/10.1155/2021/9961172

Research Article

A Secure Environment Using a New Lightweight AES Encryption Algorithm for E-Commerce Websites

Farah Tawfiq Abdul Hussien, Abdul Monem S. Rahma, and Hala Bahjat Abdul Wahab

Faculty of Computer Science, University of Technology, Baghdad 100001, Iraq

Correspondence should be addressed to Farah Tawfiq Abdul Hussien; farah.t.alhilo@uotechnology.edu.iq

Received 16 November 2021; Revised 6 December 2021; Accepted 13 December 2021; Published 24 December 2021

Academic Editor: Muhammad Arif

Providing security for transmitted data through the e-commerce environment requires using a fast and high secure encryption algorithm. Balancing between the speed and the security degree is a problem that many of the encryption algorithms suffer from. Increasing the security degree requires increasing the level of complexity which results in increasing encryption time. On the other hand, increasing the algorithm speed may reduce the complexity degree which affects the security level. This paper aims to design an encryption algorithm that balances time and complexity (speed and security). This is done by suggesting a security environment that depends on creating and providing an agent software to be settled into each customer device that manages the purchase and security process without customer interference. The suggested encryption algorithm is applied within this environment. Several modifications are performed on the AES encryption algorithm. The AES was chosen due to its performance (security and speed), which makes it suitable for encrypting transmitted data over the Internet. These modifications involve adding preprocessing steps (padding and zigzag), eliminating Sub Byte step, and reducing the number of rounds. The experimental results showed that the suggested algorithm provides more security and speed in the encryption and decryption process. The randomness degree has increased by 29.5%. The efficiency is increased because the encryption and decryption times are reduced, as is the CPU usage. The throughput for the suggested algorithm is increased by 10% for the encryption process and is increased by 9.3% for the decryption process.

1. Introduction

The huge amount of transmitted data over the e-commerce systems makes them exposed to different types of attacks [1, 2]. Therefore, providing security for these applications becomes an important issue [3, 4]. Different types of approaches have been used for this purpose; one of them is encryption algorithm [5, 6]. Providing security over the Internet requires using a fast and high secure encryption algorithm [7–9]. Creating a fast and high secure encryption algorithm requires to balance between speed and complexity due to the inverse relationship between them [10–13]. Increasing the security of any encryption algorithm requires increasing the degree of complexity through adding some additional processes, complex operations, increasing the number of rounds, and so on [14–17]. This will increase the encryption process time which leads to reduce the algorithm speed [18–20]. On the other hand, increasing the encryption process speed requires reducing complexity degree which in turn affects the security level [3, 21]. For e-commerce applications, the balancing between speed and security level is an important issue that must be considered [22].

AES is one of the most powerful encryption algorithms which is used to provide security over the Internet [23]. However, the AES has a limitation; that is, the huge calculations may reduce the algorithm speed [24]. Because of its simplicity and effectiveness, AES is one of the most widely used encryption algorithms [25–27]. However, compared to other algorithms [28–31], it consumes more computing power. Add Round Key, Sub Bytes, Shift Rows, and Mix Columns are the four transformations used in AES, and the Mix Columns transformation has the highest computational

burden of the four. There are two arithmetic operations in Mix Columns: multiplication and addition [32–35]. Because of the complicated mathematical processes that require computing resources in a software implementation of AES [36–38], it is a costly transformation that slows down the encryption process [39].

This paper suggests a modified AES encryption algorithm that aims to solve the problem of balancing between the speed and complexity; this is done by employing several operations as a preprocessing step before starting encryption (zigzag and padding), removing Sub Byte operation, and reducing the number of rounds.

The main contributions of the proposed environment and algorithm are as follows:

(1) To increase the security degree for the fixed form using a padding and zigzag pattern as preprocessing to increase the record form's character separation, which increases confusion and diffusion.

(2) To reduce encryption time by reducing the total number of AES rounds and eliminating some of the operations (Sub Bytes) for each round to consist of only three operations, except the last round consists of two operations.

(3) The algorithm performs a new form of shift columns instead of shift rows to increase the confusion and diffusion degree. The structure of the plaintext and the suggested security framework consider the fixed structure of the transmitted data and, thus, provide the required degree of security according to this fact.

(4) Creating a secure environment by providing a software agent that is settled into the customer's device. This agent is responsible for purchasing and security management without interfering with the customer.

The rest structure of this study is organized as follows. Section 2 presents related works. Section 3 presents materials and methods. Section 4 describes the results and discussion. Finally, the conclusion of the proposed approach is concluded in Section 5.

2. Related Works

In today's resource-constrained situations, the emphasis is shifting toward lightweight cryptographic algorithms. Many lightweight cryptographic algorithms have been created, as well as existing methods that have been tweaked to accommodate resource constraints.

Reference [40] discussed data security and compression using the advanced encryption standard (AES). They proposed increasing the number of rounds (Nr.) of the AES algorithm's encryption and decryption processes to 16, which increased the system's security. The initial key has been generated from the Polybius square. The encryption process undergoes the Sub Bytes, Shift Rows, Mix Columns, and Add Round Key operations. This article is based on enhancing security by increasing the number of rounds, which takes more time to calculate (time-consuming). There are no changes to the original work of the AES only increasing the number of rounds to increase complexity which increases security and at the same time increasing execution time.

This initiative, led by [41], focused on data security and compression using advanced encryption standards (AES). In our project, we increase the number of rounds (Nr) in the AES algorithm's encryption and decryption processes to 16, resulting in increased system security. This article is based on enhancing security by increasing the number of rounds, which takes more time to calculate (time-consuming). There are no changes to the original work of the AES only increasing the number of rounds to increase complexity which increases security and at the same time increasing execution time.

Reference [42] established a redesigned scheme for the encryption/decryption method by changing the Mix Columns stage. The goal of the new method is to use IV vectors, which are based on a real random number generator, to enhance the speed of the encryption/decryption process while retaining the design complexity. Such a system keeps the suggested scheme's security level as high as feasible. The Mix Columns step is replaced by an XOR operation between the input state and a random vector named IV. Then, the algorithm is executed in 16 rounds. Permutation does not offer a great deal of complexity. The complexity is reduced so much which affects the security that is reduced in turn. So, this paper increased encryption speed but reduced the security.

In [16], the advanced encryption standard is changed in the study to solve its high computing demand, which is caused by the complicated mathematical processes in Mix Columns transformation, which slows down the encryption process. Because bit permutation is simple to perform and does not require any sophisticated mathematical computation, the updated AES utilized it to replace the Mix Columns transformation in AES. The encryption time is lowered in this study. Furthermore, the complexity is reduced too much. Pit permutation is used to increase the encryption speed, but it reduced the complexity too much because the complexity of the AES depends on the Mix Column.

In the cipher round, new primitive operations, such as exclusive OR and modulo arithmetic, were added to address the poor diffusion rate in the early rounds, according to [20]. The key scheduling technique was also enhanced using byte substitution and round constant addition. To assess diffusion and confusion properties, the modified AES was compared to the regular AES using the avalanche effect and frequency test. The difficulty in this study is based on increased computations, which resulted in a longer encryption time than normal AES. The AES algorithm itself is not modified, but the key scheduling technique is made more complex which produced more complexity.

The paper in [43] used the "advanced encryption standard" (AES) algorithm and the flower pollination algorithm; this study proposes a novel method for generating the key (FPA). Modified AES is the name given to this combination (MAES). This method starts with a 128-bit

plain string as its input. This text has been converted to encrypted text. The "S-Box" generation is dependent on the key generation (substitution box). The FPA is used to generate the keys for the planned task. This procedure is done to build the keys in such a way that the S-difficulties Box's are increased. This improves the security of the proposed work for data transmission over the Internet. Then, encryption is done. The next step is decryption. Finally, at the receiver's end, the 128-bit plaintext is obtained. In this paper, the AES has not modified itself, but the technique of generating the encryption key is changed depending on the flower pollination algorithm, which consumes additional time and increases only the S-Box complexity, not the entire algorithm. Another study by [44] proposed and implemented an enhanced modification for the advanced encryption standard (AES) algorithm using an additional key generated using a linear feedback shift register (LFSR), which provides an efficient technique for pseudo-random number generation, as well as a reduction in the number of rounds. The algorithm complexity depends on key generation using LFSR. No additional randomness is shown. There are no modifications on the AES algorithm but only change in the key scheduling and generating. This study [45] proposes a secured modified advanced encryption standard algorithm that reduces the number of rounds in the advanced encryption standard (AES) to 14 to reduce encryption and decryption process time while also enhancing data security. In this study, the encryption time is reduced, but the complexity is reduced too much. It is obvious from the previous papers that all of them failed to achieve the balance between speed and security.

3. Materials and Methods

The e-commerce system has witnessed huge extensions in recent years due to the massive and various Internet technologies. This in turn led to great expansions in the size and type of transmitted data across the Internet. Some of the data contain sensitive information that may be exposed to different types of attacks, especially payment information. Therefore, security must be provided for the transmitted data. As mentioned before, this is about the e-commerce environment, where the transmitted data are characterized with the following features:

(1) The information contains financials (from which it gains importance); therefore, it must be protected against any possible intruders and attacks.

(2) Transmission security is a responsibility of the e-commerce system, and the security framework is created and managed by the e-commerce system. This is the reason for using symmetric encryption.

(3) The transmitted data have a fixed structure.

The transmitted data are arranged and packed into a record called a record form, which is described in Table 1. The e-commerce website generates a secure environment for data transmission depending on an agent structure that is responsible for two tasks: purchase management and encryption management. An agent is settled into a customer's device by his agreement to manage the purchase process and provide security without customer interference. The proposed agent can be described in Table 1.

Table 1: The structure of form records used for e-commerce environment transformation.

UserId	AgentId	ProductId	Quantity	Address	Date	Time

The e-commerce website generates a secure environment for data transmission depending on an agent structure that is responsible for two tasks: purchase management and encryption management. An agent is settled into a customer's device by his agreement to manage the purchase process and provide security without customer interference. The proposed agent is shown in Figure 1.

This agent is responsible for the data transmission management between the customer's device and the e-commerce website. This means that the record form is generated and encrypted by the agent and then sent to the commercial website. These operations are managed and achieved under the agent's control and according to the e-commerce site policies to provide the required security. The encryption process is performed according to the proposed encryption algorithm.

The proposed algorithm, which is called lightweight AES, is used to transfer data between customers and e-commerce systems over the Internet. It is used to transfer purchase information (not payment) to prevent any manipulation that can be done by an intruder during transmission. The AES algorithm is usually used for encrypting data transmission due to its secrecy, complexity, strength, and performance. However, it struggles with huge calculations. Reducing these calculations requires a long time and increases performance and security without reducing the algorithm efficiency. A modification has been made to the standard algorithm, which will be explained in the following sections, but first, there will be some preliminary steps before starting the encryption process.

3.1. The Plaintext Contents. The plaintext that will be encrypted is a record that is referred to as a form record, as shown previously in Table 1 in Section 3.

This contains the purchase information that will be sent from the customer's computer to the e-commerce site after being encrypted by an agent that is inserted into the customer's computer according to his agreement. The whole process of encryption can be described in Algorithm 1.

The input for Algorithm 1 is the order form which contains the details of the customer purchase order. This order contains UserId, AgentId, ProductId, Quantity, Address, Date, and Time. The plaintext is a sequence of characters (letters and numbers) that are converted to hexadecimal because it is the typical representation to be processed in AES processes.

3.2. Encryption Process. Before starting the encryption process, there are two steps, which are called preliminary steps. These steps involve the padding process and zigzag

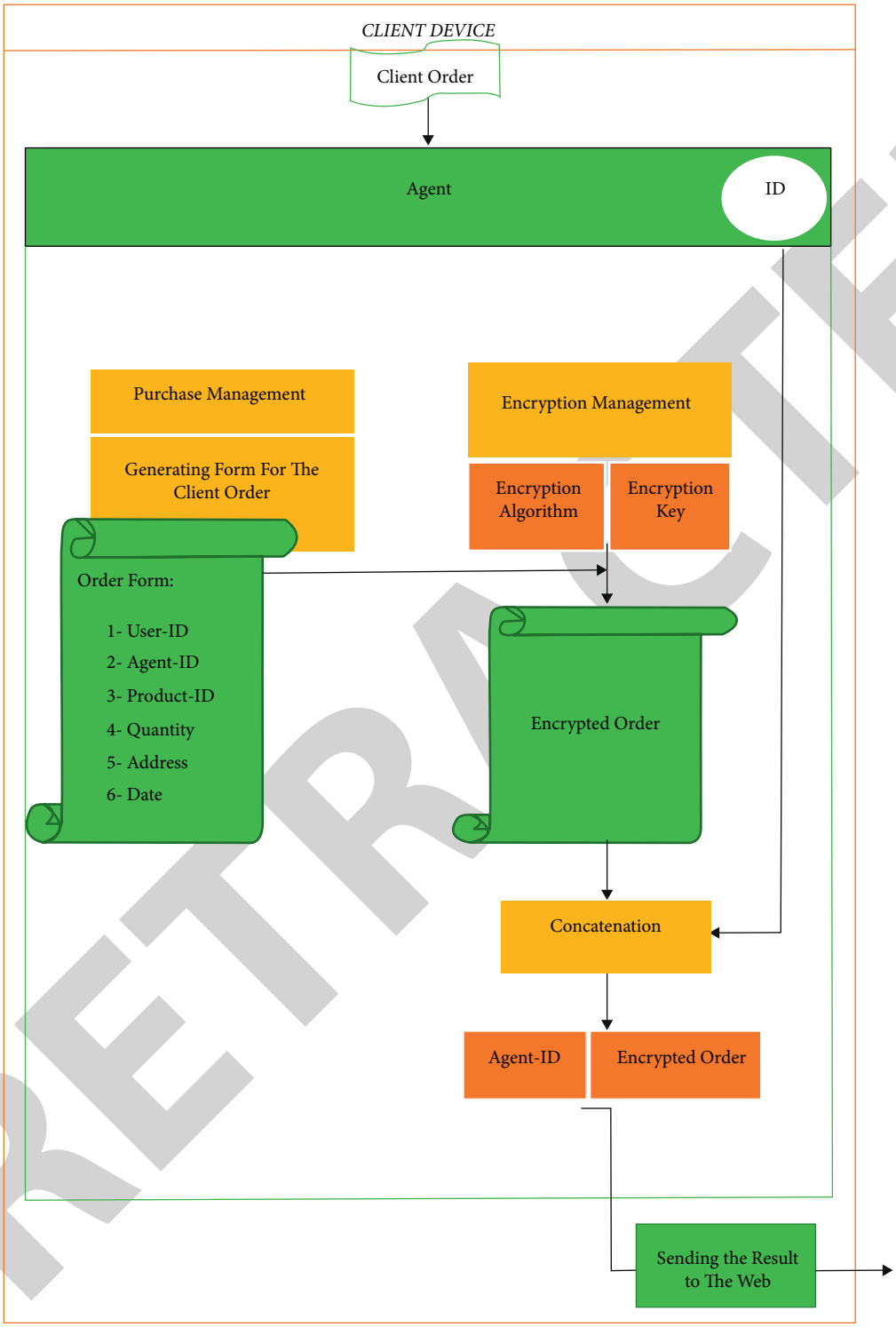

Figure 1: The proposed software agent structure and ingredients.

```
Input: Plaintext
Output: Ciphertext
Start
Read the message to be encrypted
Perform padding
Perform zigzag
Perform lightweight AES
End
```

Algorithm 1: The proposed lightweight encryption algorithm.

algorithm, which are performed on the sender side. These operations are considered preprocessing steps that reduce statistical relations among the string character before encryption.

3.2.1. Padding. This is the first process that is applied to the purchase order which is mentioned previously. The padding step aims to ensure that the string length is equal to 16, and it is multiple to be suitable for encryption because the message will be converted into a matrix of $4 * 4$ bytes.

This step can be described by Algorithm 2.

For example, suppose that we have the following string: "How are you today?"

Here, the string length is 14, and two characters need to be completed to reach the length of 16. Thus, a counter is used to specify the required number. Two characters are concatenated as expressed in the algorithm. The first one is "2," the second one is "1," and the result will be "How are you today 21." Now, the string length is 16 characters and is suitable for the next step, the zigzag.

The complexity of the padding step is $((2^8)^n)^L$, where 2^8 is the length of each character, n represents the plaintext length, and L represents the number of times of repeating the padding process for each plaintext.

3.2.2. Zigzag Pattern. The padded string is used as input for this step which is represented as a matrix of size $4 * 4$ of bytes.

To increase confusion and diffusion, a zigzag pattern is applied, as shown in Figure 2.

The zigzag pattern can be described as a rearrangement of the characters inside the string to break the statistical relations among them. This pattern is used only one time before the encryption to compensate for the elimination of the substitution step (Sub Bytes) inside the modified AES algorithm, as will be described later.

The zigzag pattern can be described in Algorithm 3.

The output of this step is a matrix of $4 * 4$ size after performing zigzag for the whole plaintext (purchase order). The result is suitable to be encrypted by the AES algorithm. The complexity for the zigzag operation is $((2^8)^{16})^L$, where 2^8 represents the length of each byte in the matrix of the zigzag, 16 represents the total number of cells in the matrix to perform zigzag on, and L represents the number of time of repeating the zigzag operation.

3.3. The Modified Encryption Algorithm. In modified AES, to reduce the execution time and the calculation time, several changes are made. First, the total number of rounds is reduced to 6 rounds. Inside each round, there are only three operations: Shift Column, Mix Column, and Add Round Key (except the final round, which has only two steps: Mix Column and Add Round Key). The Mix Column and Round Key are added in the same manner in standard AES. Mix Column operation costs a huge amount of time for the calculations, which is the most important operation that provides complexity and security. Reducing the number of rounds reduces the total time required to complete the encryption without affecting the security degree of the algorithm. Additionally, eliminating the substitution (Sub Byte) operation will save more time without affecting the AES performance. However, the zigzag method is used to provide confusion and diffusion because performing the encryption rearranges the characters of the text, but it will be performed only once before starting the AES operations, which means that it will not cost too much time. Performing the encryption process in this order using these steps provides a fast and secure encryption algorithm that is suitable for securely transforming the information over the Internet. Providing security and a fast processing time is the main goal of this paper, which is discussed in the experimental results.

3.3.1. The Shift Column Step. The shift row step is replaced by a shifting column to make it more difficult for a hacker to predict the manner of operations being performed.

Shift column is performed in the same manner as to shift row, but it is performing on columns instead of rows with some modification as described in Figure 3.

Each cell consists of bytes, so it is a matrix of $4 * 4$ of bytes. The first three columns are shifted in the same direction, while the last column is shifted in the reverse direction. This manner of shifting provides more confusion and diffusion. After the shift column process, the matrix will be as shown in Figure 4:

The shift column step can be expressed in Algorithm 4.

The complexity of the shift column operation complexity is $(2^5)^4$, where 2^5 represents the length of a complete column (number of bits to be shifted for each column) because each shift is performed for the whole column at a time, and 4 represents the number of columns to be shifted.

Input: initial message (order form as string) as matrix of $4 * 4$ of bytes
Output: Expanded message (the padded order form) as matrix of $4 * 4$ of bytes
Start
Read message (M)
Compute message length (N)
If $N < 16$ then
The number of added characters (P) = 16 – N
For $I = P$ down to 1
$M = M$ concatenate I
End

Algorithm 2: Padding.

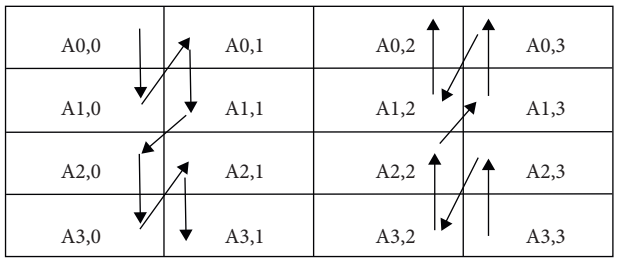

Figure 2: The initial zigzag pattern.

Figure 5 describes the details of the round in modified lightweight AES.

3.4. The Decryption Process. On the receiver side, the decryption process will be performed in reverse order using the same key as described in Algorithm 5.

The decryption process is the same as encryption but in a reverse manner, as described in Figure 6.

In the decryption process, there must be an inverse process for the mix column step and an inverse process for the shift column step, which are performed in the same manner in a different order.

3.4.1. Inverse Shift Column. The inverse shift column is performed in the same manner as the initial shift column but in reverse order, as shown in Figure 7, and by using the resulting matrix shown in Figure 4.

After performing the inverse shift column, the matrix will be returned to its original order, as shown before in Figure 3. Algorithm 6 represents the inverse of the shift column step.

3.4.2. Inverse Zigzag. Inverse zigzag is performed in the same manner as initial zigzag but in the reverse order, as shown in Figure 8.

The inverse zigzag algorithm can be described as follows in Algorithm 7.

4. Results and Discussions

Experimental results are used to prove the modified algorithm performance. These criteria involve the NIST test, encryption and decryption time, memory usage on file encryption, and file decryption. The files were encrypted to analyze the performance of the modified AES algorithm. During the experiments, different sizes of text files were tested for ten trials to get the average encryption time and CPU usage of the standard AES and modified AES. The standard and modified AES algorithms were both written in PHP Laravel Framework and simulated on Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz, 2.40 GHz with 8 GB RAM and 64-bit operating system, x64-based processor, Windows 10 Pro.

4.1. NIST Test Suite. NIST is the most widely used test for utilizing encryption algorithms. Therefore, it is used here to compare the standard AES and the new block cipher algorithm. Specifically, three tests are used for comparison: approximate entropy, run test, and linear complexity. These tests provided randomness measures for the encrypted test resulting from both standard AES and the new algorithm. The results are shown in Table 2.

The results showed that the new method produced more randomness than the standard AES. It is shown that as average the modified AES is increased by 0.209125 in approximate entropy, 0.14425 in run test, and 0.29325 in linear complexity to the standard AES.

4.2. Encryption and Decryption. Encryption and decryption time analysis is an important feature to measure the encryption algorithm performance. Different file sizes are used to measure the execution time for both the encryption and decryption steps and then compared with the standard AES. The results in Table 3 and Figures 9 and 10 show that the new algorithm is faster. The main goal of the paper is to provide a faster encryption algorithm for encrypting and decrypting data that will be transformed over the Internet.

On average, the encryption time of lightweight AES is less than the standard AES by 2858 milliseconds, while the decryption process is less by 3225 milliseconds.

4.3. Memory Space Utilization. Analyzing CPU memory using different file sizes shows that the lightweight AES uses less memory than the standard AES during the encryption

```
Input: Expanded message (EM) as matrix of 4 * 4 of bytes
Output: Zigzagged EM as matrix of 4 * 4 of bytes
Start
//For the first two column
While (i − 2) > 0 do
I = 3, J = 0
a(i, j) = a(i − 1, j)
a(i − 1, j) = a(i, j − 1)
a(i, j − 1) = a(i − 1, j − 1)
a(i − 1, j − 1) = a(i − 2, j)
i−=2
end while
//For the last two column
Count = 0
While (i + 2 ≤ 3) do
I = 0, j = 2
a(i, j) = a(i + 1, j)
a(i + 1, j) = a(I, j + 1)
a(I, j + 1) = a(i + 1, j + 1)
a(i + 1, j + 1) = a(i + 2, j)
i+=2
end while
End
```

Algorithm 3: Zigzag.

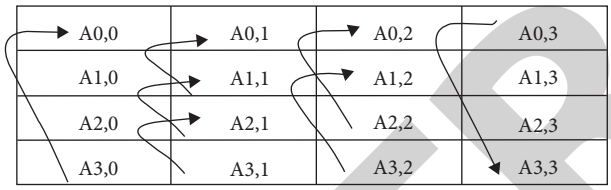

Figure 3: The initial shift column pattern.

A3,0	A1,1	A2,2	A1,3
A0,0	A2,1	A3,2	A2,3
A1,0	A3,1	A0,2	A3,3
A2,0	A0,1	A1,2	A0,3

Figure 4: The state matrix after shift column.

process. The analysis of memory usage is shown in Figure 11 and Table 4.

The CPU utilization is increased in the modified AES by 1297620 as average to the standard AES.

In addition, the memory space that is used during the decryption process in lightweight AES is less than that used by the standard AES. This is shown in Figure 12 and Table 5. The previous results showed that the lightweight AES is better at utilizing CPUs than the standard AES.

Table 6 shows the result of the avalanche effect of the standard and the modified AES. The tests were carried out by altering one bit of plaintext, either the last, first or middle bit. Although the avalanche effect of an encryption technique is dependent not only on the complexity of the algorithm but also on the key and plaintext, the modified AES created a stronger avalanche effect than the conventional AES, based on the results. The security level of the method is improved by a high avalanche effect. The results of the avalanche test result comparison between the standard and the modified AES are shown in Table 6.

It is obvious that the CPU utilization is increased in the modified AES by 961560 as average to the standard AES.

4.4. Avalanche Effect. The avalanche effect is a feature of encryption algorithms in which a change in one bit of plaintext causes several bits of the ciphertext to change. The avalanche effect is computed as follows:

$$\text{avalanche effect} = \frac{\text{no. of bits flipped in the ciher text}}{\text{no. of bits in the cipher text}}. \quad (1)$$

4.5. Comparison Analyses. A comparison analysis is shown in Table 7 to characterize the features of the suggested system using several criteria to compare with the previous works. The features which are used in Table 7 are as follows:

(1) Randomness which is taken from the NIST test

(2) Speed which indicates the encryption and decryption process speed

(3) CPU utilization refers to the memory space which is used during the encryption-decryption process

(4) Application refers to the application in which the algorithm is implemented

Input: state matrix as matrix of $4 * 4$ of bytes
Output: shifted state matrix as matrix of $4 * 4$ of bytes
$a[0, 0], a[0, 1], a[0, 2], a[0, 3] = a[3, 0], a[0, 0], a[1, 0], a[2, 0]$
$a[0, 1], a[1, 1], a[2, 1], a[3, 1] = a[1, 1], a[2, 1], a[3, 1], a[0, 1]$
$a[0, 2], a[1, 2], a[2, 2], a[3, 2] = a[2, 2], a[3, 2], a[0, 2], a[1, 2]$
$a[0, 3], a[1, 3], a[2, 3], a[3, 3] = a[1, 3], a[2, 3], a[3, 3], a[0, 3]$

Algorithm 4: Shift column.

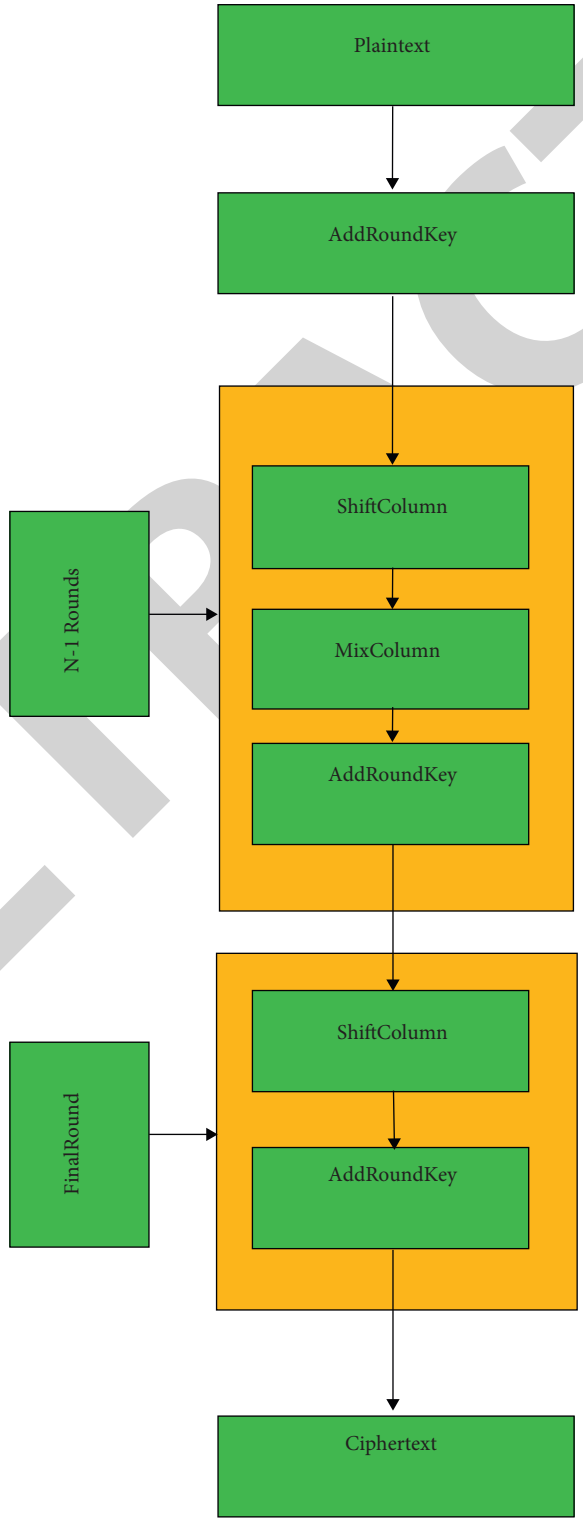

Figure 5: Lightweight AES encryption algorithm.

Input: encrypted message (the encrypted order form)
Output: decrypted message (The decrypted order form)
Start
Read encrypted message
Perform lightweight AES decryption algorithm
Perform inverse zigzag algorithm
Eliminate padding and extract the message

Algorithm 5: Proposed lightweight decryption algorithm.

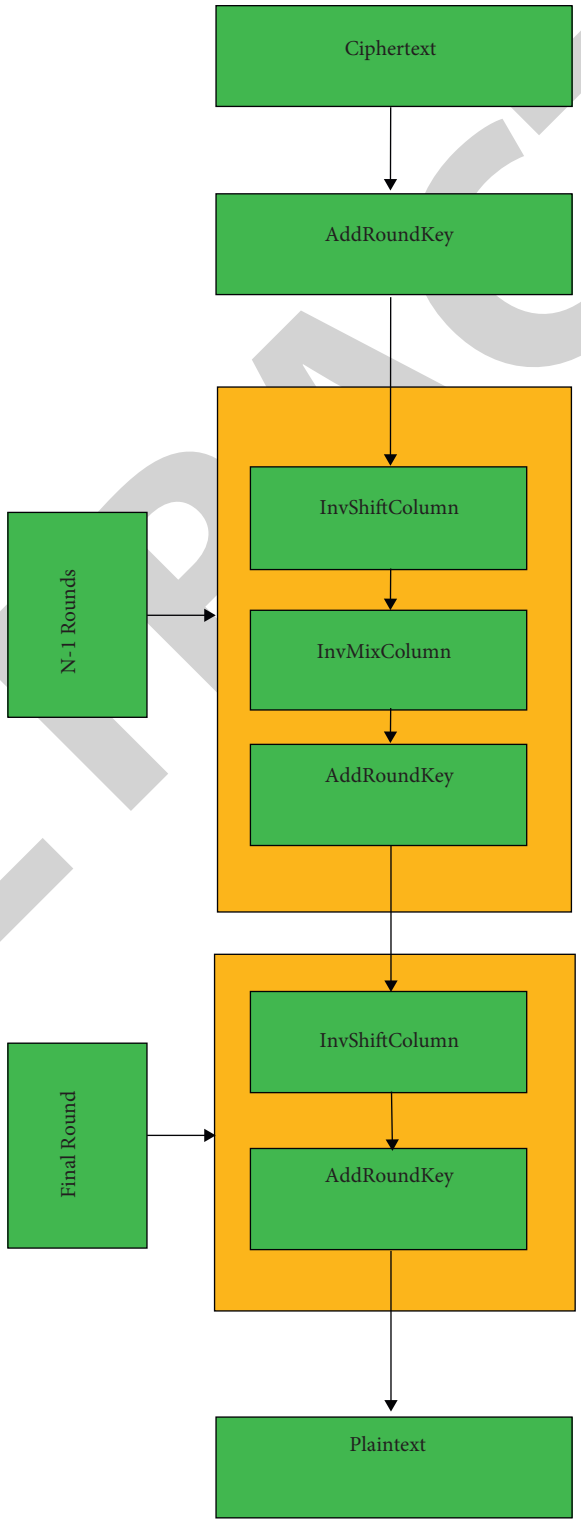

Figure 6: Lightweight AES decryption algorithm.

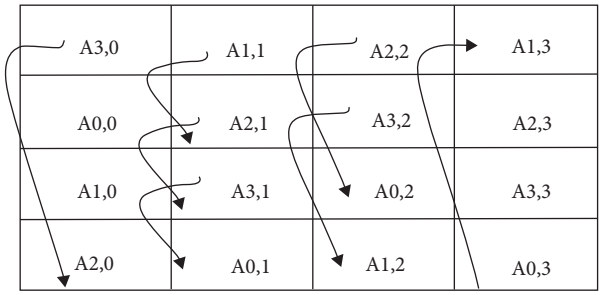

Figure 7: Inverse shift column.

Input: shifted state matrix of size $4 * 4$ of bytes
Output: state matrix of size $4 * 4$ of bytes
$a[3, 0], a[0, 0], a[1, 0], a[2, 0] = a[0, 0], a[0, 1], a[0, 2], a[0, 3]$
$a[1, 1], a[2, 1], a[3, 1], a[0, 1] = a[0, 1], a[1, 1], a[2, 1], a[3, 1]$
$a[2, 2], a[3, 2], a[0, 2], a[1, 2] = a[0, 2], a[1, 2], a[2, 2], a[3, 2]$
$a[1, 3], a[2, 3], a[3, 3], a[0, 3] = a[0, 3], a[1, 3], a[2, 3], a[3, 3]$

Algorithm 6: Inverse shift column.

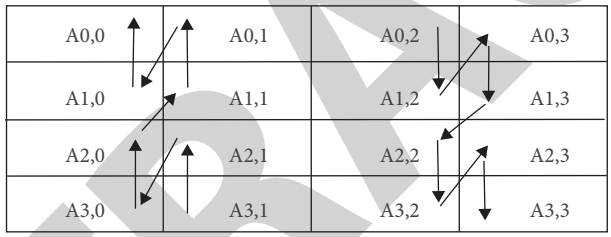

Figure 8: Inverse initial zigzag.

Input: Expanded message (EM) matrix of $4 * 4$
Output: Zigzagged EM matrix of $4 * 4$
Start
//For the first two column
While $(i + 2 \leq 3)$ do
$I = 0, j = 0$
$a(i, j) = a(i + 1, j)$
$a(i + 1, j) = a(I, j + 1)$
$a(I, j + 1) = a(i + 1, j + 1)$
$a(i + 1, j + 1) = a(i + 2, j)$
i+=2
end while
//For the last two column
$I = 3, j = 3$
While $(i - 2) > 0$ do
$I = 3, J = 0$
$a(i, j) = a(i - 1, j)$
$a(i - 1, j) = a(i, j - 1)$
$a(i, j - 1) = a(i - 1, j - 1)$
$a(i - 1, j - 1) = a(i - 2, j)$
i−=2
end while
End

Algorithm 7: Inverse zigzag.

Table 2: NIST test suite comparison between standard AES and modified AES.

Key	Standard AES			The new block cipher algorithm		
	Approximate entropy	Run test	Linear complexity	Approximate entropy	Run test	Linear complexity
kVM5HlaOSmViuDZS	0.275	0.708	0.412	0.621	0.883	0.738
nUg2Sbu5VhaNpqpU	0.591	0.269	0.891	0.95	0.869	0.896
Eio5dBioBOPJ2rY7	0.29	0.745	0.693	0.486	0.851	0.96
Ml56ROJZyEfDfPU1	0.574	0.619	0.098	0.501	0.471	0.603
DdcVWrPU3tmnrGPQ	0.641	0.619	0.098	0.858	0.982	0.862
5K3oL59ZqTsh9nBl	0.453	0.852	0.08	0.398	0.601	0.419
N38ks44Q25aOSgpD	0.013	0.684	0.99	0.316	0.932	0.863
NM2wknqgjPr5LQMh	0.51	0.902	0.654	0.89	0.963	0.921

Table 3: Analyzing the encryption and decryption phases of lightweight AES.

File size (kB)	Standard AES		Lightweight AES	
	Encryption	Decryption	Encryption	Decryption
10000	4726	5530	1260	2810
20000	5387	6328	2098	3260
30000	6100	7649	3798	4090
40000	7742	8624	4690	5140
50000	8211	9743	6030	6450

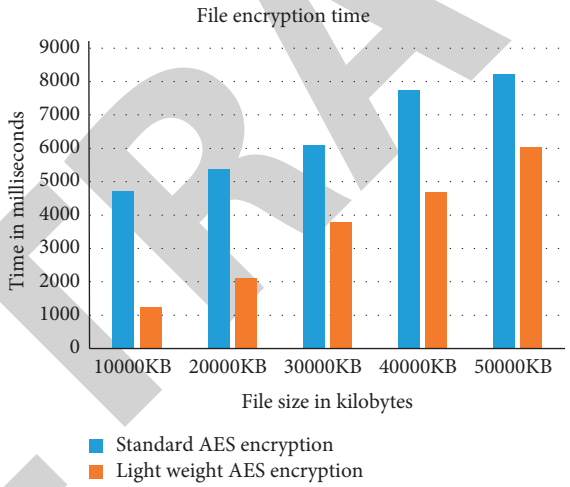

Figure 9: The encryption time for different file sizes.

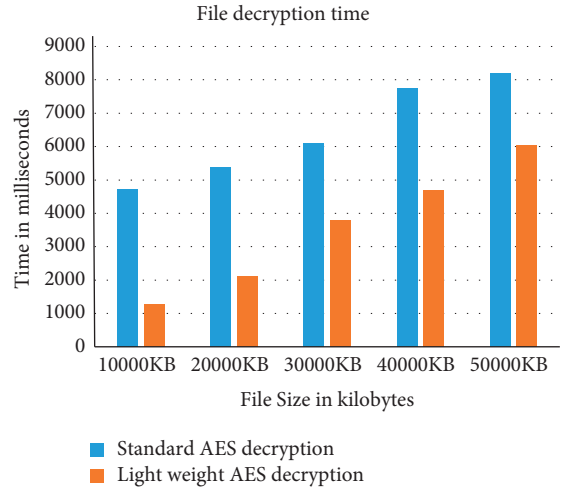

Figure 10: The decryption time for different file sizes.

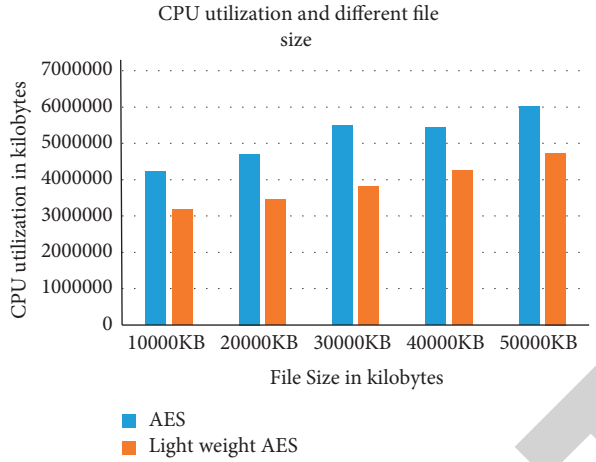

Figure 11: Memory utilization of the encryption process for different file sizes.

Table 4: Memory utilization for encrypting files of different sizes.

	CPU utilization and different file sizes				
File size	10000 kB	20000 kB	30000 kB	40000 kB	50000 kB
AES	4232445	4709445	5497823	5434024	6008559
Lightweight AES	3188566	3466432	3800123	4255876	4733201

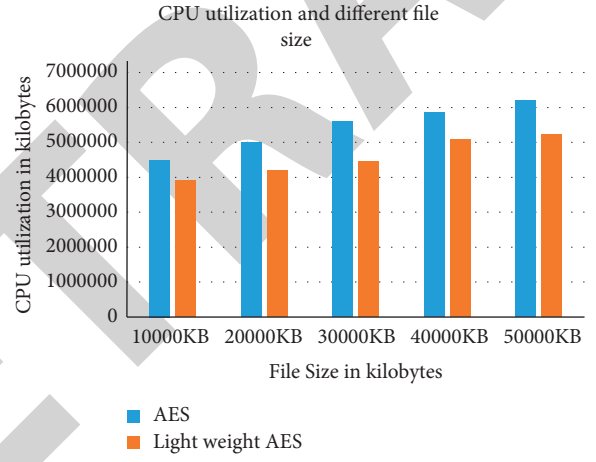

Figure 12: Memory utilization of the decryption process for different file sizes.

(5) Complexity indicates the degree of complexity to determine security level Increasing the complexity resulted in increasing the security

(6) Throughput indicates the encrypted file size per unit of time

(7) Avalanche test represents the feature of encryption algorithms in which a change in one bit of plaintext causes several bits of the ciphertext to change

The comparison analysis showed that some of the researches increase some features and reduced others which reflects the difficulty of balancing among the important features to generate an efficient algorithm. The proposed algorithm achieves the balancing among all features, especially complexity and speed which prove the efficiency of the proposed method.

The experimental results showed enhancements in the performance of the proposed method. But as seen in the discussion of the related works, reducing the number of rounds may reduce complexity and hence security which is considered as a limitation in the proposed method. This limitation is fixed by adding additional preprocessing

Table 5: Memory utilization for decrypting files of different sizes.

	CPU utilization and different file sizes				
File size	10000 kB	20000 kB	30000 kB	40000 kB	50000 kB
AES	4499130	4991030	5610998	5849331	6214445
Lightweight AES	3911033	4188970	4456991	5076798	5223344

Table 6: Avalanche effect result.

Plaintext	The ciphertext (AES)	%	The ciphertext (modified AES)	%
1110001111111111	2jhlfv483jhsd4khgsja7op349f93jsv	46.67	158c023be5d70c50545f3d61607a860c	59.83
0110001111111111	a8hfry49cfbrjvfdsivgresd39586fjd	61	1771755582db80b309fc0457ab25d380	76
0110001111111111**1**	nvhdyr053ufjlgjfhe7g4y560yokghdg	48.93	82db80b2aa8b0cbfd7b466d309fc0457	60.59
0110001111111110	vdk3islpea98dmxl2tmnz8usuw92ort1	60	4b4a6e0642692a02802699fee75d0f25	77
1000111123456789	jdkfli83bc7wux038d7fhhfy7e383292	43.01	8cdb801eb884a4f793ddb42b6a937897	54.62
0000111123456789	sa89dgew6tfrejhddfgg093jfuryewid	57	f7f37dc2e2b9d7cbf4870c90b20b4e70	69
A1B2DDE3245BC6F**9**	3mvbeu8eu3fnvkfjrue83w22bmto9utu	42.51	4a247aed6aa93193890ef3478285ceb5	53.78
A1B2DDE3245BC6F**8**	ikv48etyos9234mncc4dclffgnbbgtr4	55	9557144b4a6e06e75d0f251fb253e980	67
9876661234567890	jd83la9dg3uc3ty9o3iqwf5mnp781lsn	40.81	ffb27b9f9e74b0781b90d2c06f51f213	59.10
9876661234567891	kg93i6vhas2qr492kdjfyur8ekfnehdu	66	91d1b68274db79ebb83bda50876854f	75
amjvtrhc**p**sjhgawl	dje21fp9woj63polmeerfsaaa234399	41.55	c4383d7fbead6f2691b5cc94ebd6efb6	60.80
amjvtrhc**q**sjhgawl	gh34lk8cmn94ls39gasjh2dkv45uv786	58	9bb1c42692a02802699fead619c975fd	77
abxhskhl**e**rabzskp	1n53sakffe324lkj287mnbzxlsepwqit	50.68	09bcce44b4a6e06e75d0f25828acf496	66.04
abxhskhl**i**rabzskp	rtiyu2349idjsndjgjb09244kaodlkwo	69	e5d70c5050dc4e81be0d9daa545f3d61	85

The avalanche test of the modified AES is increased as average by 14.328 to the standard AES as shown in Table 6.

Table 7: A comparison analysis between the proposed method and the previous works.

Papers	Randomness	Speed	CPU utilization	Application	Complexity	Throughput	Avalanche effect
[35]	Not specified	Increased	Not specified	Internet banking, account passwords, email account password	$(((2^8)^4)^{16})^{16}$	Not specified	Not specified
[36]	Not specified	Increased	Not specified	Not specified	$(((2^8)^4)^{16})^{18}$	Not specified	Not specified
[37]	Not specified	Decreased	Not specified	Not specified	$((2^8)^{16})^{10}$	Not specified	Not specified
[38]	Decreased	Decreased	Enhanced	Text and images	$((2^8)^{16})^{10}$	Decreased	Reduced
[39]	Decreased	Decreased	Enhanced	Text and images	$(((2^8)^{16})^4)^{10}$	Decreased	Increased
[40]	Not specified	Increased	Enhanced	Text and images	$(((2^8)^{16})^4)^{10}$	Decreased	Not specified
[41]	Not specified	Decreased	Not specified	Text, image, and video	$(((2^8)^{16})^4)^5$	Not specified	Not specified
[42]	Not specified	Decreased	Not specified	Text	$((2^8)^{16})^{14}$	Increased	Not specified
The proposed method	Increased	Decreased	Enhanced	Encrypting purchase order in e-commerce systems	$(((2^8)^4)^{16})^6 + ((2^8)^{16})^L + (2^8)^n)^L$	Increased	Increased

operation that increases little bit the security of the algorithm and adding more randomness for the new method. Also, settling the software agent inside the customer device occupies additional memory space, which is considered as another limitation but it is acceptable because it enhances the e-commerce site by reducing the deadlock situation for which the proposed system is suggested originally.

5. Conclusions

The large volume of data transformed over the Internet has led to a strong need to protect data from theft and manipulation, especially sensitive and financial data. Encryption is one of the most important and most common methods used to protect data from theft, but encryption algorithms struggle with some problems, including the time required for encryption, as the data transmitted over the Internet must be encrypted at an acceptable speed. This paper presents a proposal to modify the AES algorithm to reduce the time taken for encryption while maintaining the level of complexity necessary to protect the data. In this algorithm, it was found that the Sub Byte operation that was being executed in all rounds was canceled and replaced by the zigzag algorithm, which was used once before starting the encryption process. Since the total number of cycles is 6 cycles, each cycle consists of three operations, which are Added Round Key, Mix Column, Shift Column, except for the last cycle, which consists of only two operations (Add Round Key and Mix Column). The modified algorithm increases the confusion and diffusion by employing the

padding and zigzag patterns as preprocessing before the encryption process. Adding padding and zigzag algorithm adds more complexity to the algorithm. The performance is increased by decreasing the encryption and decryption time, which is considered a critical issue in real-time systems, such as e-commerce systems. It also requires fast processing without affecting the complexity level to support the security level. Reducing the number of rounds resulted in reducing the encryption and decryption processes which increase the modified algorithm speed. On the other hand to balance between the speed and complexity, two operations are added as preprocessing steps (padding and zigzag) before encryption to add more confusion and diffusion and add more complexity to the algorithm to keep the level of security acceptable. Experiments showed an increase in the efficiency of the algorithm in terms of reducing the encryption and decryption time, while improving the use of memory and CPU resources, maintaining the amount of complexity required to maintain data confidentiality, and increasing the avalanche percentage.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

References

[1] A. S. Hamada and A. K. Farhan, "Image encryption algorithm based on substitution principle and shuffling scheme," *Engineering Technology J.*vol. 38, no. 3, pp. 98–103, 2020.

[2] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung, "A survey on the IETF protocol suite for the Internet of Things: standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.

[3] R. Riyaldhi and A. Kurniawan, "Improvement of advanced encryption standard algorithm with shift row and S.box modification mapping in mix column," *Procedia Computer Science*, vol. 116, pp. 401–407, 2017.

[4] Prathiba and V. S. K. Bhaaskaran, "Lightweight S-box Architecture for secure internet of things," *Info*, vol. 9, no. 1, pp. 1–14, 2018.

[5] H. Ali Abdulmohsin, H. B. Abdul Wahab, and A. M. Jaber Abdul hossen, "A new proposed statistical feature extraction method in speech emotion recognition," *Computers & Electrical Engineering*, vol. 93, pp. 1–14, 2021.

[6] R. J. Toama and M. Nada Hussein, "A secure cipher for the gray images based on the shamir secret sharing scheme with discrete wavelet haar transform," journal of," *Mech. of Contin. & Math. Sci.*vol. 15, no. 6, pp. 334–351, 2020.

[7] S. Afzal, M. Yousaf, H. Afzal, N. Alharbe, and M. R. Mufti, "Cryptographic strength evaluation of key schedule algorithms," *Security and Communication Networks*, vol. 2020, pp. 1–9, 2020.

[8] K. Suhad Muhajer and S. Abdul Monem, "Rahma, "New method for improving add round key in the advanced encryption standard algorithm," *Information Security Journal*, vol. 30, no. 6, pp. 371–383, 2021.

[9] R. Saha, G. Geetha, G. Kumar, T.-h. Kim, and R. K. Aes, "RK-AES: an improved version of AES using a new key generation process with random keys," *Security and Communication Networks*, vol. 2018, pp. 1–11, Article ID 9802475, 2018.

[10] M. K. Pehlivanoğlu, M. T. Sakalli, N. Duru, and F. B. Sakalli, "The new approach of AES key schedule for lightweight block ciphers," *IOSR Journal of Computer Engineering*, vol. 19, no. 3, pp. 21–26, 2017.

[11] C. Li, F. Zhao, C. Liu, L. Lei, and J. Zhang, "A hyperchaotic color image encryption algorithm and security analysis," *Security and Communication Networks*, vol. 2019, Article ID 8132547, 8 pages, 2019.

[12] V. Heidilyn and Gamido, "Implementation of a bit permutation-based advanced encryption standard for securing text and image files," *Indones. J. of Electr. Eng. & Comp. Sci.*, vol. 19, no. 3, pp. 1596–1601, 2020.

[13] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and Advanced Encryption System," *Signal Processing*, vol. 141, pp. 217–227, 2017.

[14] O. Omoruyi, C. Okereke, K. Okokpujie, E. Noma-Osaghae, O. Okoyeigbo, and S. John, "Evaluation of the quality of an image encryption scheme," *TELKOMNIKA (Telec. Comput. Elec. Cont.*, vol. 17, no. 6, pp. 2968–2019.

[15] R. L. Quilala, A. M. Sison, and R. P. Medina, "Modified SHA-1 algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 1027–1034, 2018.

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

[17] H. V Gamido, A. M. Sison, and R. P. Medina, "Implementation of modified AES as image encryption scheme," *Indones. J. Electr. Eng. Informatics*, vol. 6, no. 3, pp. 301–308, 2018.

[18] H. Ali-Pacha, N. Hadj-Said, A. Ali-Pacha, M. Mamat, and M. A. Mohamed, "An efficient schema of a special permutation inside of each pixel of an image for its encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 2, pp. 496–503, 2018.

[19] S. Rehman, S. Q. Hussain, W. Gul, and Israr, "Characterization of advanced encryption standard (AES) for textual and image data," *Int. J. Eng. Comput. Sci.*vol. 5, pp. 18346–18349, 2016.

[20] E. M. De Los Reyes, A. M. Sison, and R. Medina, "Modified AES cipher round and key schedule," *Indones. J. Electr. Eng. Informatics*, vol. 7, no. 1, pp. 29–36, 2019.

[21] H. Talirongan, A. M. Sison, and R. P. Medina, "Modified advanced encryption standard using butterfly effect," in *Proceedings of the 2018 IEEE 10th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Control. Environ. Manag.*, IEEE, Baguio City, Philippines, Dec 2018.

[22] B. A. Khalaf, S. A. Mostafa, A. Mustapha et al., "An adaptive protection of flooding attacks model for complex network environments," *Security and Communication Networks*, vol. 2021, Article ID 5542919, 17 pages, 2021.

[23] E. M. De Los Reyes, A. M. Sison, and R. P. Medina, "File encryption based on reduced-round AES with revised round keys and key schedule," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 897–905, 2019.

[24] M. Elhoseny, G. Ramirez-Gonzalez, O. M. Abu-Elnasr, S. A. Shawkat, N. Arunkumar, and A. Farouk, "Secure medical data transmission model for IoT-based healthcare Systems," *IEEE Access*, vol. 6, pp. 20596–20608, 2018.

[25] X. Zhou, Y. Ma, Q. Zhang, M. A. Mohammed, and R. Damaševičius, "A reversible watermarking system for medical color images: balancing capacity, imperceptibility, and robustness," *Electronics*, vol. 10, no. 9, Article ID 1024, 2021.

[26] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 565–576, 2011.

[27] H. Hu Xiong, "Cost-effective scalable and anonymous c remote authentication protocol," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2327–2339, 2014.

[28] M. N. Fadhil Ibraheem, "Proposed hybrid-encryption system for multicast network," *Engineering Technology J.*vol. 28, no. 24, pp. 7027–7036, 2010.

[29] M. Poongodi, M. Malviya, M. Hamdi et al., *5G Based Blockchain Network for Authentic and Ethical Keyword Search Engine*, IET Communications, U.K, 2021.

[30] A. M. S. Rahma, A. M. J. Abdul Hossen, and O. A. Dawood, "Public key cipher with signature based on diffie-hellman and the magic square problem," *Engineering Technology J.*vol. 34, no. 1, pp. 1–15, 2016.

[31] A. M. S. Rahma and A. M. Abbas, "A modified matrices approach in advanced encryption standard algorithm," *Engineering Technology J.*vol. 37, no. 3, pp. 86–91, 2019.

[32] A. Lakhan, M. Abed Mohammed, S. Kadry, K. Hameed Abdulkareem, F. Taha AL-Dhief, and C.-H. Hsu, "Federated learning enables intelligent reflecting surface in fog-cloud enabled cellular network," *PeerJ Computer Science*, vol. 7, p. e758, 2021.

[33] I. A. A. Abdul-Jabbar and S. M. Kadhim, "Copyright protection service for mobile images," *Engineering Technology J.*vol. 34, no. 4, pp. 444–450, 2016.

[34] Y. Ye, N. Wu, X. Zhang, L. Dong, and F. Zhou, "An optimized design for compact masked AES S-box based on composite field and common subexpression elimination algorithm," *Journal of Circuits, Systems, and Computers*, vol. 27, no. 11, pp. 1–11, 2018.

[35] M. E. Hameed, M. M. Ibrahim, and N. AbdManap, "Review on improvement of advanced encryption standard (AES) algorithm based on time execution, differential cryptanalysis and level of security," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1, pp. 139–145, 2018.

[36] G. Kalpana, P. V. Kumar, S. Aljawarneh, and R. V. Krishnaiah, "Shifted adaption homomorphism encryption for mobile and cloud learning," *Computers & Electrical Engineering*, vol. 65, pp. 178–195, 2018.

[37] X. Zhang, S.-H. Seo, and C. Wang, "A lightweight encryption method for privacy protection in surveillance videos," *IEEE Access*, vol. 6, pp. 18074–18087, 2018.

[38] M. Atheer and A. Al-Albbassi, *Database Encryption System Based on AES Algorithm and 3-D Box*, Ph.D. dissertation, Dept. Comp. Sci., University of Technology, Baghdad, Iraq, 2020.

[39] M. N. Al-Mhiqani, R. Ahmad, Z. Z. Abidin et al., "A new intelligent multilayer framework for insider threat detection," *Computers & Electrical Engineering*, vol. 97, Article ID 107597, 2021.

[40] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," *Optik*, vol. 127, no. 4, pp. 2341–2345, 2016.

[41] B. Nageswara Rao, D. Tejaswi, K. Amrutha Varshini, K. Phani Shankar, and B. Prasanth, "Design of modified AES algorithm for data security," *Int. J. For Tech. Res. In Eng.*vol. 4, no. 8, pp. 1289–1292, 2017.

[42] M. A. eltatar, *Modified Advanced Encryption Standard Algorithm for Reliable Real-Time Communications*, master dissertation, Dept. of Eng., The Islamic University–Gaza, Gaza, 2017.

[43] M. Indrasena Reddy and A. P Siva Kumar, "A secure approach for data transmission in computer Networks using modified advanced encryption standard algorithm," *J. Mech. Cont.& Math. Sci., Special Issue*, no. 3, pp. 14–28, 2019.

[44] H. H. Ali and S. H. Shaker, "Modified Advanced Encryption Standard algorithm for fast transmitted data protection," in *Proceedings of the 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020), IOP Conf. Series: Materials Science and Engineering*, vol. 928, pp. 1–11, Nov 2020.

[45] N. A. Mohd Ariffin and A. Y. Ahmed Ashawesh, "Enhanced AES algorithm based on 14 rounds in securing data and minimizing processing time," *J. of Phy.: Conference Series*, vol. 1793, pp. 1–9, Article ID 012066, 2021.