Modified Rijndael Algorithm for Resource-Constrained IoT-Based Wireless Sensor Networks

Mohammad Sirajuddin[1,*] and Dr. B. Sateesh Kumar[2]

[1]Research scholar, Department of Computer Science Engineering, Jawaharlal Nehru Technological University (JNTU), Hyderabad, Telangana 500085, India
[2]Professor, Department of Computer Science Engineering, JNTUH University College of Engineering Jagitial, Telangana 505501, India

Abstract

INTRODUCTION: IoT devices are small, have limited battery and computing capabilities, and cannot tolerate complex encryption. So there is a need for a lightweight cryptographic scheme that can perform well in resource constraint environments.
OBJECTIVES: Many encryption algorithms struggle to balance security and complexity. Increasing the level of protection necessitates increasing complexity, which leads to higher encryption and decryption times. The goal of this article is to offer a novel encryption method, the Modified Rijndael Algorithm, which balances security and complexity.
METHODS: The classic Rijndael encryption technique has been modified to make it lighter and more secure, suitable for resource-constrained IoT devices. These modifications include adding the novel pre-processing step Crossover based on the two-point crossover operation of genetic algorithms, removing the Sub Byte step, changing the shift column phase, and decreasing the rounds.
RESULTS: The proposed scheme provides speed and high security, increasing efficiency and randomness while reducing encryption and decryption times by 21.06% and CPU consumption by 13.2% compared to the traditional method.
CONCLUSION: The proposed algorithm reduces encryption time and maintains the complexity needed to protect the data. It is lighter and more secure, making it suitable for IoT devices with limited resources.

Keywords: modified Rijndael algorithm , MRA, lightweight cryptography, IoT security

Received on 05 October 2022, accepted on 01 January 2023, published on 07 August 2023

doi: 10.4108/eetsis.2748

1. Introduction

The IoT plays a significant role in various aspects of our daily life. The Internet of Things environment usually includes smart devices, healthcare equipment, buildings, automobiles and other elements. Sensors, software, integrated circuits, and actuators are incorporated in IoT devices and sense the information from the surroundings. Sensed data utilised by administrators control various equipment. Data security has become a compulsory feature for IoT-based WSNs in recent years due to the various security threats causing problems to the IoT equipment; the importance of security increased as IoT devices have limited resources. Data captured from various locations send to the cloud via the internet, so secure communication is required because of the high sensitivity of IoT applications like the military, smart home, and health care [1].

Data security is considered one of the most crucial challenges in IoT; it is an essential requirement, particularly for data-driven processes and transactions. Before transferring information over a network, data encryption is essential to provide security and privacy to the sensed information.

[*]Corresponding author. Email:mohdsiraj569@gmail.com

IoT devices are getting smaller and more efficient in terms of performance. The number of devices linked to the internet has dramatically increased because of technological advancements in hardware and software. This number is anticipated to rise significantly over the coming years as new communication technologies are introduced. One of the greatest concerns is data security; protecting information over this growing network with dynamic topologies is challenging. Without addressing security issues, future IoT technologies might compromise user data privacy. Since sensor nodes are bound by their limited resources, it is challenging to use complex security mechanisms like Firewalls, Intrusion Detection Systems. In the IoT environment, protecting gathered information is very crucial. To ensure security, communication between these devices should be encrypted, many encryption mechanisms are available to secure the data, but these algorithms are very sophisticated and require a lot of resources.

WSN uses small batteries to power the sensors with limited memory and computational capabilities. Cryptographic algorithm-based solutions solve IoT security challenges; nevertheless, these strategies must be designed and implemented in line with the restricted resources of IoT devices. Conventional encryption techniques may need more energy to encrypt data, significantly shortening the component lifespan. As a result, a lightweight encryption scheme became necessary to protect data. Lightweight cryptography provides adequate security with low resource consumption [2]. Lightweight cryptography may be applied more successfully in small, low-energy, low-footprint devices such as RFID tags and sensors.

Nowadays, Researchers are showing interest in lightweight cryptography to address security problems of resource-constrained IoT devices. Currently, the trend in resource-constrained applications is moving toward lightweight cryptographic algorithms. Numerous efficient cryptographic algorithms have been created, and even existing techniques have been modified for environments with limited resources. The main security risk in WSNs is that the devices used to collect data from the real world may be the subject of cyberattacks. This issue is exacerbated further by the heterogeneous nature, diverse resources, various node performance needs, and, more importantly, security attacks.

This paper proposes a block cypher algorithm that balances speed and complexity. Several changes are made to the conventional Rijndael Algorithm to decrease execution and computation time and suit energy-constrained IoT-based Wireless sensor networks. Crossover is the first step applied to plain text. Crossover creates confusion and diffusion by rearranging the characters. The modified Rijndael algorithm reduces the overall number of rounds to six. Each round contains only three steps: Shift Columns, MixColumn, and AddRoundKey (the S-Box step is eliminated), except the last step, which contains only two phases: MixColumn and AddRoundKey.

IoT Security Challenges

Large numbers of wireless sensor nodes (equipment) are deployed in the IoT environment; their design and architecture provide numerous benefits and drawbacks. Attackers are particularly interested in these weaknesses and attack the environment using them [2]. Figure 1 shows IoT Security challenges and reasons why attackers chose specific devices as targets.

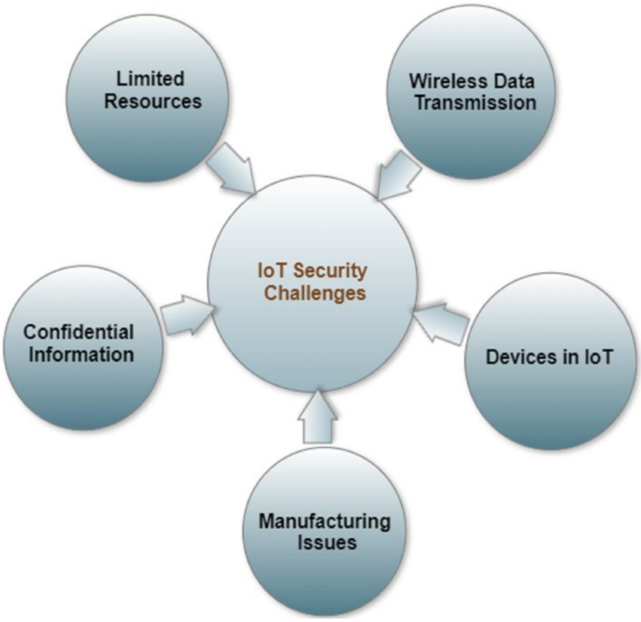

Figure 1. IoT Security Challenges

Devices in IoT

The IoT environment uses a vast number of devices spread across a broad surface area [2]. Therefore, this scenario appeals to attackers since the more devices, the greater the number of entrance points captured.

Limited resources

Because IoT devices are small and have limited battery and computing capabilities, security solutions such as firewalls and antivirus software cannot be installed. As a result of this predicament, attackers are likely to target these devices.

Wireless data transmission

When using wireless networks, security is a significant concern. A wireless network that is not installed correctly or maintained might pose serious security risks. Situations may deteriorate if the network is not adequately protected. They require a wireless adaptor, which risks hacking by allowing hackers easy access to the network [3].

Confidential information

IoT devices collect and store confidential information. Indeed, maintaining confidential information in a system attracts attackers' attention and increases the number of attacks on this data, for example, military data, health data, and data gathered by a home automation system [4].

Manufacturer issues

Manufacturers' backdoor releases and a lack of timely software updates may pose severe security risks in IoT-based WSNs.

2. Lightweight cryptography

The significant interconnection of devices and massive data on the wireless Internet of Things (IoT) has made information vulnerable to several cyberattacks. Cryptographic algorithms are used to ensure information integrity and guarantee secrecy. However, because of its small size, limited processing capabilities, memory, and battery resources, it is challenging to use resource-intensive conventional cryptographic techniques for information security. In this circumstance, developing lightweight security strategies for IoT becomes necessary [3]. Many cryptographic systems are employed to provide security services and are classed as symmetric or asymmetric. However, because IoT devices are resource-restricted, most classical cryptosystems cannot be employed to protect IoT environments. To ensure safe data transmission in IoT networks, the development of efficient and lightweight cryptographic algorithms has become a problem. Lightweight encryption should use minimal energy, computation, and memory in cyber-physical systems. It should also provide a good balance of security and complexity.

Data security for confidentiality and integrity is provided when encryption is employed on sensor equipment, and this might be a potent attack deterrent. Since lightweight cryptography serves this purpose, secure encryption may be employed on devices with few resources. The size is the first factor that determines whether a device can implement it. When it comes to energy-harvesting devices, power is extremely crucial, whereas battery-powered devices prioritise power consumption. For devices that transmit huge amounts of data, such as cameras or vibration sensors, high throughput is required, yet a short latency is crucial for real-time applications. The size becomes the benchmark for both the power and the lightness of the encryption technique because the power is so reliant on the hardware, such as the size of the circuits or the CPU being used. Due to the execution time, which depends on processing speed, the number of calculations used to estimate processing speed is used as a measure of how light something is. The capacity to perform parallel processing has a significant impact on throughput.

2.1. Rijndael Algorithm

Rijndael is a block encryption algorithm designated as the Advanced Encryption Standard by the NIST. Vincent Rijmen and Daemen Joan, two Belgian cryptographers, invented Rijndael. It succeeds the Data Encryption Standard (DES). For encrypting sensitive (unclassified) American federal data, NIST has chosen Rijndael as the standard symmetric key encryption algorithm. The decision was made after thoroughly examining Rijndael's security and efficiency properties. Rijndael's architecture is based on simple mathematical ideas such as algebra for matrix manipulation and finite field mathematics [2]. Rijndael produces data blocks from one-dimensional 8-bit byte arrays as input. The plaintext is input first, and then the state bytes are mapped to it. The cypher key is also an 8-bit byte array with one dimension. Rijndael is a block cypher that is iterated. It is possible to encrypt or decrypt a data block by iterating a specific transformation (a round function).

Encyphering with Rijndael

The Rijndael encryption is a block cypher with interactivity. As a result, it consists of modifications to encrypt or decrypt the data. Mixing subkeys with the data block is the initial step of Rijndael encryption and decryption [5]. An extra step is used to protect from cryptanalysis. Do an AddRoundKey step on its own to decode a data block in Rijndael, then the normal rounds, and finally, the final transformation step without the MixColumn step to decode a data block in Rijndael. The following stages define the encryption itself:

- Round Key stage.
- N-1 Rounds.
- Last round.

Where N denotes the number of rounds to be completed, the round transformation is divided into layers. Four transformation stages are used to create these layers. A non-linear byte substitution is performed via the ByteSub step. A cyclic shift is the Shift Row transformation [4]. The MixColumn phase follows, in which the State's columns are modulo multiplied with a fixed polynomial and represented as polynomials over a limited field. The Add Round Key step is completed last.

The important objective of this article is to create an efficient encryption method suitable for resource-constrained IoT-enabled wireless sensor networks, requiring fewer resources and taking less time to encrypt and decrypt the data while maintaining a decent balance between complexity and security. The proposed encryption technique uses the traditional Rijndael algorithm to make it appropriate for small devices with limited battery power. The Modified Rijndael algorithm is suitable for securing confidential data such as health care information, financial data, and data obtained from various sensors deployed in the environment such as battlefields, forests, underwater, and smart homes.

3. Literature survey

The Secure Low Power Communication approach is presented in [6], a highly secure yet low-power communication system for LoRaWAN that reduces end-device ta encryption power by minimising AES encryption cycles. To increase security and AES encryption process simplification while decreasing resource utilisation, the SeLPC introduces a D-Box update mechanism and an encryption key. When compared to standard AES, the SeLPC can lower encryption time by up to 26.2%, according to the research. In addition, the SeLPC can survive three sorts of attacks: known-key, replay, and eavesdropping making it effective in LoRaWAN IoT scenarios.

The article [7] presents AES encryption and decryption as low-cost countermeasures. This solution employed temporal redundancy and altered the AES round design. The suggested architecture divides the round into three parts, with two pipeline registers added in between to increase fault coverage. The proposed technique can identify 99.539 % of randomly inserted faults. The overhead is relatively minimal, and the suggested technique outperforms alternative methods.

The paper [8] presented a common AES-based lightweight encryption technique. The first step was to use chaotic Boolean functions to create a cryptographically safe S-box. The Lorenz system and Hilbert curve scan pattern are used to implement the diffusion and permutation steps. The chaotic S-cryptographic box's features and NIST testing prove its strength. The method was created with extremely limited IoT devices in mind. The method is light enough to fulfil various requirements, including memory consumption, execution time, and information entropy, according to experimental data.

The study [9] offers a technique to protect sensor data; a new Lightweight Advanced Encryption Standard (LAES) depends on a mix of chaotic mechanisms, was developed. It was created with the goal of shortening encryption and decryption times. AES generates S-Box, initial permutation (IP), shifting quantities, and cryptographic keys by combining logistic maps in varying proportions. The S-Box is a vital step in AES, and to boost security, a new S-Box is constructed utilising a 1D chaotic logistic map system. IP has taken the place of shift rows, while dynamic Shift Row has taken the place of MixColumns. The new LAES, according to the study, can encode a 5MB text file within only 1.987 seconds.

In most aspects, the solution suggested in [10] is comparable to the conventional AES algorithm; however, it does not need the use of a MixColumn, as the standard AES algorithm requires. The VIVADO tool simulates and synthesises the enhanced lightweight AES 128-bit algorithm's implementation. Artix-7 and Kintex-7 FPGAs are used to analyse the algorithm's output. In terms of FPGA resources consumed and algorithm time delay, the lightweight AES approach is compared.

The article [11] described an encryption system that added a white box to AES and doubled the encryption. This method also used a white box instead of the normal AES's Substitute-Byte (S-Box). A white box is significant because it marks the moment where the entire AES encryption is separated into round functions. While doubling the AES process makes it more difficult for an attacker or malware to disrupt the network or system, it also makes it more difficult for an attacker or malware to disrupt the network or system. According to the findings, the recommended solutions can prevent DoS attacks on IoT and other small devices.

The article [12] utilises a 3-dimensional S-box to demonstrate an important scheduling approach (substitution box). The logistic map method has been implemented to improve security. The suggested method is suitable for lightweight IoT devices like smart household appliances. The study looks at how the appearance of chaos speeds up the planned key initiation before message transmission. The recommended technique is evaluated using the needed generation delay for smart-home sensor devices.

The work [13] presents a new low-cost implementation of the AES, a flexible symmetric encryption method, for edge devices with time-multiplexed designs. This research intends to develop a low-power, high-throughput AES architecture that may be employed in resource-constrained applications. The resource-sharing strategy and a modified Substitution box are used to optimise AES encryption/decryption hardware four times, resulting in a maximum operating frequency of 1.053 GHz.

The paper [14] introduces MAES (Modified Advanced Encryption Standard), a simplified variation of the Advanced Encryption Standard (AES) that satisfies the requirement. By developing a brand-new equation for building a square matrix during the affine transformation phase of MAES, a brand-new 1-dimensional Substitution Box is provided. With a packet transmission efficiency rate of roughly 18.35%, MAES is more energy-efficient than AES and suitable for environments with resource constraints.

The aim of this research [15] is to evaluate and contrast the networking performance of IoT devices using the Simon-Speck and AES encryption methods. The test's parameters include latency, throughput, memory consumption efficiency for the encryption technique, and the avalanche effect's value. According to experimental findings, the Speck algorithm performs better than the Simon and the AES algorithms in terms of memory use and communication latency. The Simon method has the greatest average avalanche effect value compared to the Speck and the AES algorithms in terms of avalanche effect values.

The Sub Byte and Shift Rows stage and the InvSubByte and InvShiftRows stage in the encryption section, as well as the InvSubByte and InvShiftRows stage in the decryption section, are modified in the paper [16] to alter the AES technique. The classical AES method was slightly modified to achieve the proposed methodology. After evaluating the outcomes, the ML-AES produced positive outcomes for obtaining improved transmission criteria. Additionally, (ML-AES) was 10% quicker than the original AES and used less memory. Additionally, for the ML-AES and the classic AES, the avalanche used to compute diffusion properties are (52.1% and (51.8%, respectively).

In an 8-bit AVR environment, the study [17] offered a novel column-wise implementation of AES. It effectively structured registers to compactly integrate key operations of AES and resolved the FACE-LIGHT proposal's non-constant time problem. At all security settings, the suggested program in ECB mode outperforms the previous best implementation by up to 10.2%. Additionally, the improved approach may be used in many other operating modes, including CBC, CTR, CFB, OFB, CCM, and GCM. We use the suggested AES-CTR mode and give an optimised CTR DRBG implementation. We got a performance gain of up to 10.04% for CTR DRBG over the present implementation using the suggested approach with AES-128.

4. Proposed system

The Proposed Encryption mechanism was created for resource-constrained IoT devices, and it is light enough to fulfil various requirements, including memory consumption, execution time, and information entropy. The proposed security technique Modified Rijndael Algorithm (MRA) aims to develop an encipher mechanism that balances complexity and speed. Several changes are made to the conventional Rijndael Algorithm to decrease execution and computation time. Mapping is the first procedure applied to plain text by rearranging the characters; Crossover creates confusion and diffusion. MRA reduces the overall number of rounds to six. Each round contains only three steps: Shift Columns, MixColumn, and AddRoundKey (the S-Box step is eliminated), excluding the last round, which contains only two operations: MixColumn and Add-RoundKey. Like the conventional Rijndael Algorithm, The MixColumn and RoundKey functions are implemented similarly. The Mix Column action requires a significant time for calculations, but it is the most critical operation in terms of complexity and security. The input plain text is first applied to a Crossover phase, which can be thought of as rearranging the characters of plain text to disrupt the analytical relationships between them. Crossover is performed only once before encryption to accommodate the S-box round of the conventional Rijndael algorithm.

4.1. Encryption Process

At the beginning of data encryption, a preliminary step known as Crossover must be accomplished on the sender's end. This operation is regarded as a pre-processing step because it minimises the statistical relationships between the string characters before the encryption process begins. Figure 2 shows the encryption using MRA.

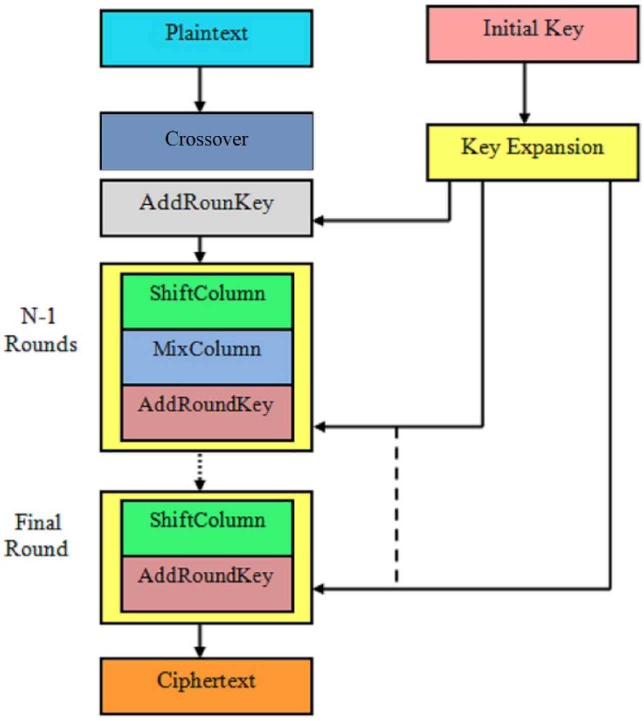

Figure 2. Modified Rijndael Algorithm (MRA) Encryption

Modified Rijndael Algorithm

1. Divide the plain text into 16-byte blocks and transform each block into a 4*4 state matrix.
2. For each block of size 4*4, perform the following steps.
3. Apply pre-processing step Crossover on the block of size 4*4.
4. Receive the Key from the Key Expansion module and perform the AddRoundKey operation with a state of size 4*4.
5. For i in the range (1,6)
    Apply ShiftColumn operation.
    Apply MixColumn operation.
    Perform AddRoundKey by using another key from the Key-Expansion Module.
6. Perform ShiftColoum and AddRoundkey steps using another key from the Key-Expansion Module.
7. To acquire ciphertext, save the encrypted data block and repeat steps 3, 4, 5, and 6 on the remaining data blocks.

Pseudo code for Modified Rijndael Algorithm

```
MRA_Encryption( plaintext, key)
{
   blocks= divideIntoBlocks(plaintext);
   For Each block of size 4*4 in plaintext do     {
       Crossover(block);
       AddRoundKey(RounKey[0], block);
       For i=1 to 6 do            {
            ShiftColumn(block);
            MixColumn(block)
            AddRoundKey(RounKey[..], block);
          }
      ShiftColoum(block)
     AddRoundKey(RounKey[5], block);
   }
  ciphertext=reassemble(block);
  return ciphertext;
}
```

Crossover

Crossover can be considered a reorganisation of the characters inside the pattern to avoid the quantitative relationship. The crossover was used just once before encryption to compensate for the absence of Sub Bytes in the modified Rijndael algorithm. The output of the Crossover step is a matrix of 4 ∗ 4 sizes after conducting mapping of characters from the plaintext, and the output is appropriate to the MRA. The Crossover phase takes the plain text of 16-byte blocks and converts each block into a 4*4 state matrix. On this 4*4 state matrix, Mapping will be used first, followed by a two-point crossover operation of a genetic algorithm to increase the complexity. In Mapping, input bits are mapped based on a specific pattern, as shown in figure 3, which creates a zigzag among the bits and makes prediction very difficult. Additionally, a two-point crossover from genetic algorithm operation will be used, making the process very complex and unpredictable to intruders. The 4*4 state matrix will be transformed into two 8-bit parent chromosomes for the two-point crossover procedure. Following the two-point crossing procedure indicated in figure 4, offspring chromosomes will be created. New child chromosomes are utilised to create a 4*4 state matrix, which will be passed into the following step.

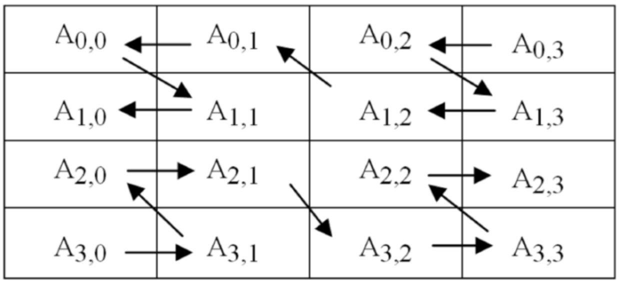

Figure 3. Mapping

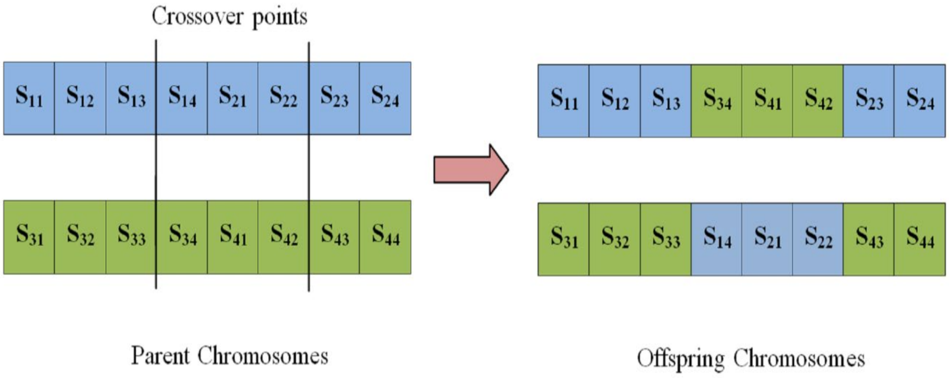
Crossover points
S_{11} S_{12} S_{13} S_{14} S_{21} S_{22} S_{23} S_{24}
S_{31} S_{32} S_{33} S_{34} S_{41} S_{42} S_{43} S_{44}
Parent Chromosomes
S_{11} S_{12} S_{13} S_{34} S_{41} S_{42} S_{23} S_{24}
S_{31} S_{32} S_{33} S_{14} S_{21} S_{22} S_{43} S_{44}
Offspring Chromosomes

Figure 4. Two Point Crossover

The Shift Columns Step

The shifting column substitutes the shift row step to make it harder for hackers to predict. The shift column works like a row shift; it works on columns instead of rows, with some modifications. Each cell contains a block of data of 4*4 size; the odd columns are moved in one way, but the even columns are moved in the other direction, the first and second columns are moved by one-bit positions, and the third and fourth columns are moved by two-bit positions as shown in Figure 5.1 causing more confusion and dispersion. Figure 5. 2. Shows data block of size 4*4 after shift column step. Every shift is done for a single column at a time; the shift column operation's complexity is $(2^5)^4$, where 2^5 denotes the length of an entire column and four denotes the number of columns to be moved. It's because each shift is executed for the entire column at once. Figure 5 shows the Shift Column Step.

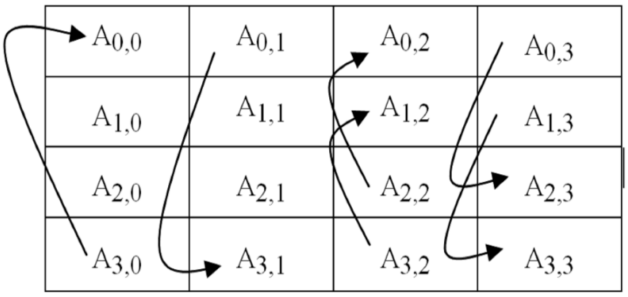

Figure 5.1. Odd columns are moved in one way and Even columns are moved in another way

$A_{3,0}$	$A_{1,1}$	$A_{2,2}$	$A_{2,3}$
$A_{0,0}$	$A_{2,1}$	$A_{3,2}$	$A_{3,3}$
$A_{1,0}$	$A_{3,1}$	$A_{0,2}$	$A_{0,3}$
$A_{2,0}$	$A_{0,1}$	$A_{1,2}$	$A_{1,3}$

Figure 5.2. Data block after shifting columns

Figure 5. Shift Column Step

4.2 Decryption Process

Reverse-ShiftColumn

In Figure 6, the reverse ShiftColumn is performed as same as the ShiftColumn, but in the opposite order. Figure 6.1 shows the shifting of odd columns in one way and even columns in another way, just opposite of the shift column step, and figure 6.2 shows the resultant matrix after application. After conducting the inverse ShiftColumn, the matrix Reverse-ShiftColoumn step is in its original order.

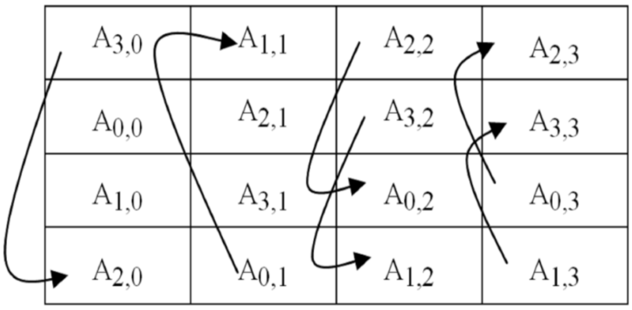

Figure 6.1. Columns moved in the opposite order of the Shift Column step

$A_{0,0}$	$A_{0,1}$	$A_{0,2}$	$A_{0,3}$
$A_{1,0}$	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
$A_{2,0}$	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$
$A_{3,0}$	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$

Figure 6.2. Data block after shifting columns in the opposite direction

Figure 6. Reverse-ShiftColumn

Inverse-Crossover

Inverse Crossover is carried out similarly to the initial Crossover but in the opposite order. Initially, the reverse-Crossover will be performed, and then the inverse crossover will be applied. All remaining operations are performed in reverse order as the traditional Rijndael algorithm.

The fundamental goal of this framework is to provide an effective encryption technique appropriate for wireless sensor networks with limited resources, which uses fewer resources and takes less time to encrypt and decryption data while retaining a reasonable level of complexity and security. For compact devices with constrained battery life, the Modified Rijndael algorithm is suitable. The Modified Rijndael algorithm is appropriate for protecting sensitive data related to health care and finances, as well as information gathered from various sensors placed throughout the environment, including those used in smart homes, forests, underwater environments, and battlegrounds.

5. Results and discussions

The Performance of Modified Rijndael is demonstrated using experimental results. The performance of MRA is measured by the amount of time it takes to encrypt and decrypt files and the number of resources utilised to encrypt and decode files. These results compare the traditional

Rijndael algorithm against the new lightweight Modified Rijndael algorithm.

5.1. Encryption and Decryption times

The principal aim of this article is to offer a faster and more lightweight encryption technique for IoT-based Wireless Sensor Networks. Time analysis of encryption and decryption is crucial for evaluating encryption algorithm performance. Encryption and decryption times are measured with different file sizes compared to the regular Rijndael Algorithm. According to the results, the new algorithm is faster than the old one. The proposed method offers speed and improved security in encryption and decryption, increasing efficiency and randomness while reducing encryption and decryption times by 21.06% compared to the traditional method, and Figure 7 shows the Encryption Times of Rijndael and MRA. Table 1 shows the encryption and decryption times of the Modified Rijndael algorithm.

Table 1. Encryption and Decryption times of Modified Rijndael Algorithm.

File Size	Standard Rijndael		Modified Rijndael	
	Encryption (ms)	Decryption (ms)	Encryption (ms)	Decryption (ms)
10000KB	4725	5731	1370	2880
20000KB	5389	6327	1998	3750
30000KB	6101	7648	3851	4020
40000KB	7743	8625	4726	3854
50000KB	8212	9744	6009	5998

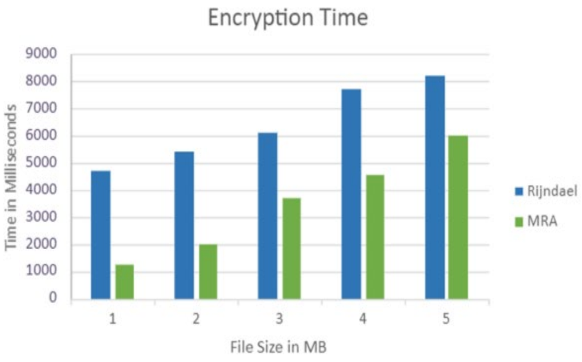

Figure 7. Encryption Times of Rijndael and MRA

In the Proposed methodology Modified Rijndael Algorithm, the data encryption time is reduced to a certain level, as shown in Figures 8. and 9, which is better than the traditional algorithms like Rijndael, DES, and Triple DES. The proposed encryption strategy uses a reduced number of rounds and requires fewer computations which make this strategy consume less time for encryption and suitable for resource constraint environments like IoT-based wireless sensor networks.

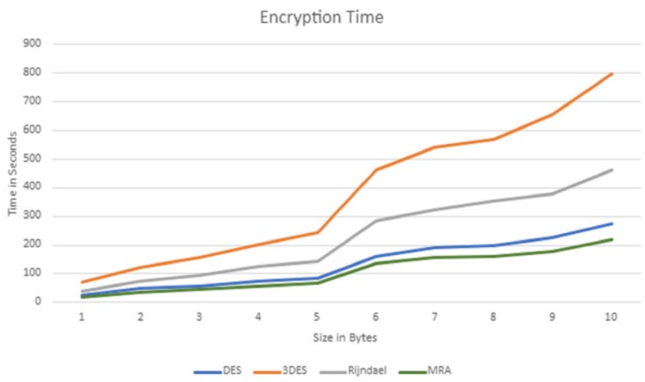

Figure 8. Encryption Time comparison of various cryptographic standards with MRA

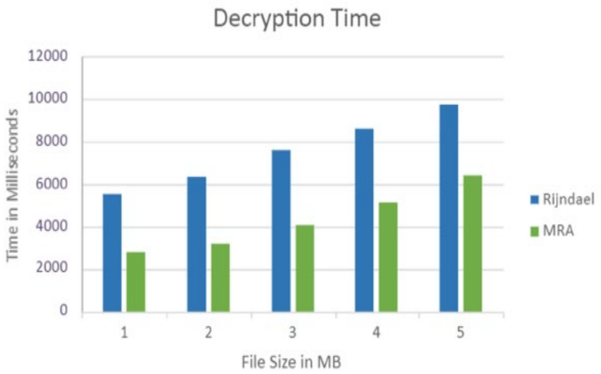

Figure 9. Decryption Times of Rijndael and MRA

In comparison to algorithms like Rijndael, DES, and Triple DES, the data decryption time is decreased to a specific level in the proposed methodology Modified Rijndael Algorithm, as illustrated in Figure 10. The suggested encryption method utilises fewer rounds and requires fewer calculations, which reduces the time necessary for encryption and makes it ideal for situations with limited resources, such as wireless sensor networks based on the Internet of Things.

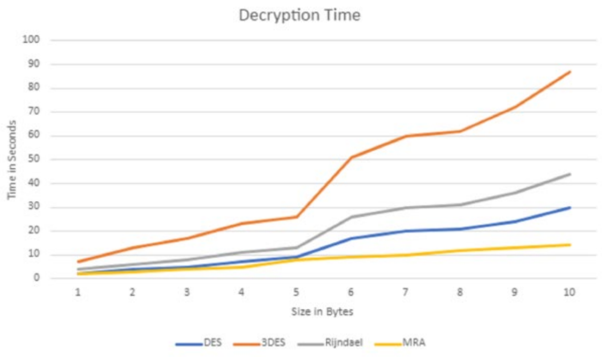

Figure10. Decryption Time comparison of various cryptographic standards with MRA

5.2. Resource Utilisation

When comparing resource consumption across different file sizes, the lightweight MRA algorithm uses fewer IoT device resources during encryption and decryption than the classic Rijndael algorithm. Furthermore, the memory space and CPU utilisation needed by lightweight MRA during the encryption process is lesser than that of the regular Rijndael algorithm. The proposed technique provides speed and higher security in encryption and decryption, increasing efficiency and randomness while reducing CPU consumption by 13.2% compared to the traditional method. Table 2 shows the CPU utilisation of Rijndael and the Modified Rijndael Algorithm. Figure 11 represents the CPU utilisation comparison of Rijndael and MRA.

Table 2. CPU utilisation of MRA

File Size	Standard Rijndael	Modified Rijndael
10000KB	4232445	3235345
20000KB	4709445	3636349
30000KB	5497823	3900101
40000KB	5434024	4199988
50000KB	6008559	4822568

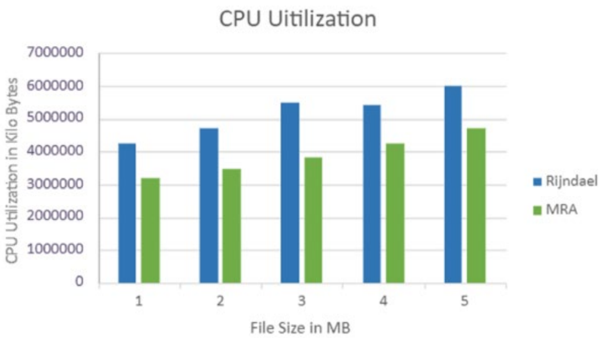

Figure 11. CPU Utilisation of Rijndael and MRA

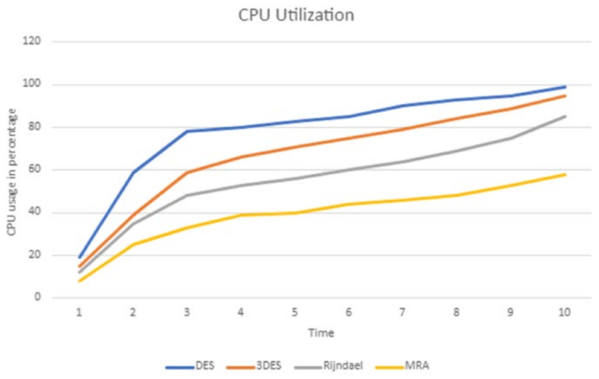

Figure 12.CPU utilisation comparison of various cryptographic standards with MRA

The proposed strategy MRA required memory was analysed while encrypting and decrypting the data blocks depicted as seen in Figure 12, and the proposed scheme uses less amount of memory compared to the classical Rijndael, DES, and 3DES; it proves that the proposed MRA is better than the classical approaches in resource constraint environments.

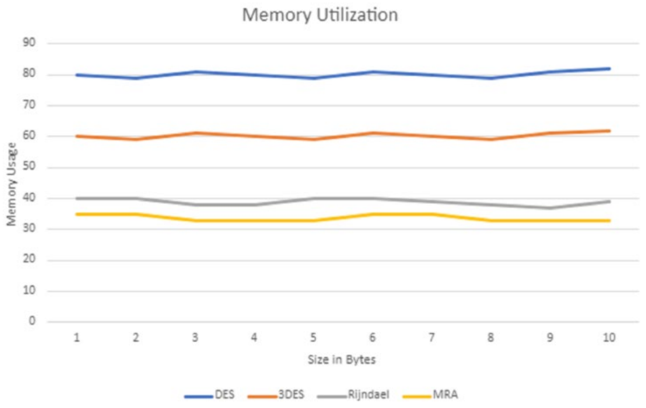

Figure 13. Memory utilisation comparison of various cryptographic standards with MRA

When encrypting and decrypting the data block packets, the suggested technique MRA's power consumption was examined and shown as seen in Figure 13. Compared to the traditional Rijndael, DES, and 3DES, the suggested approach consumes less amount of power. It demonstrates that the suggested MRA is superior to the traditional techniques in contexts with resource constraints.

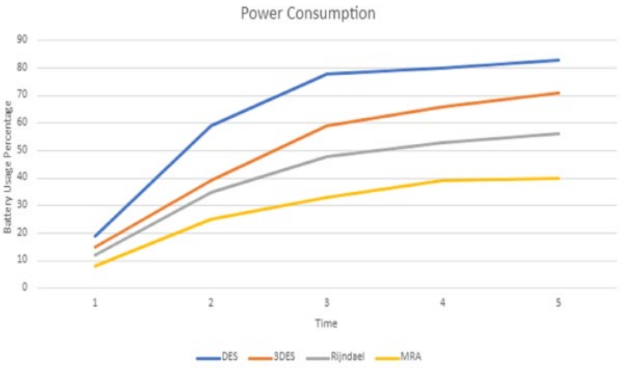

Figure 14. Power consumption comparison of various cryptographic standards with MRA

While encrypting and decrypting the data blocks, the proposed approach uses resources in the form of CPU utilisation, was examined and shown in Figure 14. Compared to the traditional Rijndael, DES, and 3DES, the suggested approach consumes a reasonable amount of CPU. It demonstrates that in contexts with resource constraints, the suggested MRA is superior to the traditional techniques.

5.3. Avalanche Effect

In block encryption, the avalanche effect, which describes how a slight modification in the data incorporates a considerable change (avalanche) in the output, is an essential requirement in encryption. More avalanche means the hacker would have more difficulty cracking the message because of its effect on diffusion computation. A block cypher or cryptographic hash function has inadequate randomisation if the avalanche effect is not present to a substantial extent. In this case, a cryptanalyst may anticipate the input using just the output. The algorithm could be broken in part or whole by this. Therefore, from the perspective of the creator of the cryptographic method or device, the avalanche effect is a desirable circumstance [18]. One of the main design goals is to build a cypher or hash that exhibits a significant avalanche effect [19]. Because of this property, tiny changes can spread quickly through an algorithm's iterations, ensuring that before the algorithm runs out of time, every bit of the output must rely on every bit of the input[20]. The proposed method's performance was improved based on the experimental findings. However, as discussed in the linked studies, decreasing rounds can diminish difficulty, and hence privacy is a drawback of the suggested technique. This issue is addressed with the addition of the Crossover pre-processing procedure, which improves the algorithm's security and adds more unpredictability to the new technique.

Table 3. Avalanche effect result

Plain Text	Ciphertext (Rijndael)	Avalanche	Cyphertext (MRA)	Avalanche
0000111123456789	sa89dgew6tfrejhddfgg093jfuryewid	57%	f7f37dc2e2b9d7cbf4870c90b20b4e70	69%
A1B2DDE3245BC6F8	ikv48etyos9234mncc4dclffgnbbgtr4	55%	9557144b4a6e06e75d0f251fb253e980	67%
amjvtrhcqsjhgawl	gh34lk8cmn94ls39gasjh2dkv45uv709	56%	1d1b68274db79ebb83bda50876854f	72%
5876661234567891	kg93i6vhas2qr492kdjfyur8ekfnehdu	59%	09bcce44b4a6e06e75d0f25828acf496	71%

Table 3 shows the experimental findings demonstrating improvements in the suggested approach's performance[21]. However, as can be observed from the discussion of comparable studies, lowering the number of rounds may lower complexity and hence security, which is regarded as a drawback in the suggested strategy. By including an extra pre-processing step, this constraint is resolved. A comparison study revealed that some studies increased certain aspects while decreasing others, which highlights how difficult it is to strike a balance between the crucial elements when creating an effective algorithm [22]. The proposed algorithm successfully balances all aspects, mainly complexity and speed, demonstrating the effectiveness of the suggested approach.

6. Conclusion

The large amount of data collected by IoT devices must be protected against theft and manipulation. Enciphering is a significant and widely used way of protecting information from intruders. However, enciphering techniques face challenges such as time and resource consumption, exacerbated in resource-constrained IoT situations. This article proposes a modification to the Rijndael algorithm to minimise encryption time while keeping the amount of complexity required to secure data. These modifications include adding the pre-processing step Crossover, removing the Sub Byte step, changing the shift column phase, and decreasing the number of rounds.

Since there are six rounds in total, each cycle has three operations: AddRoundKey, MixColumn, and ShiftColumn, except for the last round, which has only two steps (AddRoundKey and MixColumn). The new mechanism creates confusion and dispersion by using mapping as the pre-processing step before the encryption process. Crossover increases the algorithm's complexity while decreasing the number of stages saves encryption time. As a result, performance is improved by reducing encryption and decryption time, a crucial problem in a resource-constrained IoT-based Wireless sensor network environment. The proposed Modified Rijndael Algorithm balances security and complexity, is lighter and more secure, and is appropriate for IoT devices with scared resources. Experiments revealed that the algorithm's effectiveness improved in reducing encryption and decryption times while lowering memory and CPU use while retaining the complexity needed to ensure data privacy and boosting the avalanche.

Acknowledgements.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflict of Interest

The authors declared that they have no conflicts of interest in this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Availability of data and material

Not applicable

Code availability

Not applicable

Author contributions

The corresponding author claims the major contribution of the paper, including formulation, analysis and editing. The co-author provides guidance to verify the analysis result and manuscript editing.

Compliance with ethical standards

This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

References

[1] Harn L, Hsu CF, Xia Z, He Z. Lightweight Aggregated Data Encryption for Wireless Sensor Networks (WSNs). IEEE Sensors Letters. 2021 Mar 3;5(4):1-4.

[2] Abdul Hussien FT, Rahma AM, Abdul Wahab HB. A secure environment using a new lightweight AES encryption algorithm for e-commerce websites. Security and Communication Networks. 2021 Dec 24;2021.

[3] Sirajuddin M, Kumar BS. Efficient and Secured Route Management Scheme Against Security Attacks in Wireless Sensor Networks. In 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) 2021 Aug 4 (pp. 1045-1051). IEEE.

[4] Krivtsov V, Birkinshaw S, Olive V, Lomax J, Christie D, Arthur S. Multiple benefits of blue-green infrastructure and the reduction of environmental risks: Case study of ecosystem services provided by a suds pond. InCivil Engineering for Disaster Risk Reduction 2022 (pp. 247-262). Springer, Singapore.

[5] Fadhil MS, Farhan AK, Fadhil MN. A lightweight AES Algorithm Implementation for Secure IoT Environment. Iraqi Journal of Science. 2021 Aug 31:2759-70.

[6] Tsai KL, Huang YL, Leu FY, You I, Huang YL, Tsai CH. AES-128 based secure low power communication for LoRaWAN IoT environments. IEEE Access. 2018 Jul 5;6:45325-34.

[7] Bedoui M, Mestiri H, Bouallegue B, Hamdi B, Machhout M. An improvement of both security and reliability for AES implementations. Journal of King Saud University-Computer and Information Sciences. 2022 Jan 13.

[8] Alshammari BM, Guesmi R, Guesmi T, Alsaif H, Alzamil A. Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic S-box. Symmetry. 2021 Jan 13;13(1):129.

[9] Fadhil MS, Farhan AK, Fadhil MN, Al-Saidi NM. A New Lightweight AES Using a Combination of Chaotic Systems. In2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA 2020 Jul 12 (pp. 82-88). IEEE.

[10] Kumar K, Ramkumar KR, Kaur A. A lightweight AES algorithm implementation for encrypting voice messages using field programmable gate arrays. Journal of King Saud University-Computer and Information Sciences. 2020 Aug 15.

[11] Rahman Z, Yi X, Billah M, Sumi M, Anwar A. Enhancing AES Using Chaos and Logistic Map-Based Key Generation Technique for Securing IoT-Based Smart Home. Electronics. 2022 Mar 30;11(7):1083.

[12] Shanthi Rekha S, Saravanan P. Low-cost AES-128 implementation for edge devices in IoT applications. Journal of Circuits, Systems and Computers. 2019 Apr 26;28(04):1950062.

[13] Shanthi Rekha S, Saravanan P. Low-cost AES-128 implementation for edge devices in IoT applications. Journal of Circuits, Systems and Computers. 2019 Apr 26;28(04):1950062.

[14] Chowdhury AR, Mahmud J, Kamal AR, Hamid MA. MAES: Modified advanced encryption standard for resource constraint environments. In2018 IEEE Sensors Applications Symposium (SAS) 2018 Mar 12 (pp. 1-6). IEEE.

[15] Yustiarini BY, Dewanta F, Nuha HH. A Comparative Method for Securing Internet of Things (IoT) Devices: AES vs Simon-Speck Encryptions. In2022 1st International

Conference on Information System & Information Technology (ICISIT) 2022 Jul 27 (pp. 392-396). IEEE.

[16] Hammod DN. Modified Lightweight AES based on Replacement Table and Chaotic System. In2022 International Congress on Human-Computer Interaction, Optimisation and Robotic Applications (HORA) 2022 Jun 9 (pp. 1-5). IEEE.

[17] Kim Y, Seo SC. Efficient Implementation of AES and CTR_DRBG on 8-bit AVR-based Sensor Nodes. IEEE Access. 2021 Feb 16;9:30496-510.

[18] Tang C, Cheng Y, Yin J. An Optimized Algorithm of Grid Calibration in WSN Node Deployment Based on the Energy Consumption Distribution Model. JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE. 2012;9(4):1035-42

[19] Aslan B, Yavuzer Aslan F, Sakallı MT. Energy Consumption Analysis of Lightweight Cryptographic Algorithms That Can Be Used in the Security of Internet of Things Applications. Security and Communication Networks. 2020 Nov 21;2020.

[20] Assafli HT, Hashim IA. Security Enhancement of AES-CBC and its Performance Evaluation Using the Avalanche Effect. In2020 3rd International Conference on Engineering Technology and its Applications (IICETA) 2020 Sep 6 (pp. 7-11). IEEE.

[21] Salman RS, Farhan AK, Shakir A. Lightweight Modifications in the Advanced Encryption Standard (AES) for IoT Applications: A Comparative Survey. In2022 International Conference on Computer Science and Software Engineering (CSASE) 2022 Mar 15 (pp. 325-330). IEEE.

[22] Tang C, Yin J. A localization algorithm of weighted maximum likelihood estimation for wireless sensor network. Journal of Information &Computational Science. 2011 Dec;8(16):4293-300.