Journal of Advanced Zoology

ISSN: 0253-7214
*Volume **44** Issue **S-6 Year 2023** Page **1323:1332***

An Enhanced and Dynamic Key AES Algorithm for Internet of Things Data Security

Dhamodharan G

Assistant Professor PG & Research Department of Computer Science, Don Bosco College (Co-Ed), Guezou Nagar, Yelagiri Hills, Affiliated to Thiruvalluvar University.

**Corresponding author's E-mail: haidhamo@gmail.com*

Article History

Received: 06 June 2023
Revised: 05 Sept 2023
Accepted: 03 Dec 2023

Abstract

Internet of Things (IoT) applications have become ubiquitous in various aspects of daily life, including smart homes, healthcare, and other areas where human assistance is crucial. However, the valuable real-world data collected by IoT devices and transmitted over the Internet have become a prime target for numerous malicious actors and hackers. Therefore, ensuring secure communication to prevent unauthorized access to this transmitted data is of paramount importance. This research is dedicated to the development of a robust security system for IoT to protect sensor data. Traditionally, IoT systems have relied on fixed encryption keys shared between transmitters and receivers, which presented vulnerabilities as these keys could be easily compromised. In this study, we introduce an enhanced version of the AES algorithm with dynamic keys. In this modified algorithm, each cycle involves the XOR operation of four keywords with the sequence number. Each round of the AES algorithm incorporates various transformations, including DivideSwap, SubBytes, ShiftRows, MixColumns, and AddRoundKey, with the exception of the final round, which excludes the MixColumns operation. To implement and test this upgraded security approach, we utilized hardware components such as the Arduino Uno, ESP8266, and DHT11 sensors. The enhanced AES algorithm was integrated into the Arduino Uno to secure sensor data before transmission across the network. The results of our analysis demonstrate that the proposed IoT security method presents a significantly higher level of cryptographic resilience compared to the traditional AES algorithm, making it a robust solution for safeguarding IoT data.

Keywords: *DivideSwap, AES, Internet of Things, Arduino Uno, DHT11, Security*

1. Introduction

The Internet of Things (IoT) has become an integral part of our lives, harnessing interconnected devices capable of sensing, computing, and sharing data across networks and the Internet. While various encryption techniques [1] exist, they often introduce communication delays due to additional processing requirements [2]. Ensuring the security of transmitted data remains paramount, particularly because IoT sensors are typically powered by small batteries and possess limited processor, memory, and energy resources [3]. As a result, the need for lightweight encryption techniques has arisen to safeguard data effectively [4].

Lightweight cryptography aims to achieve robust security while minimizing resource utilization [5]. Several studies, such as those outlined in [5][6][7][8], have focused on optimizing common algorithms, presenting valuable insights that can benefit the IoT ecosystem. One avenue of research has explored enhancing the security of the AES algorithm by developing new S-Boxes to replace the traditional ones, given the S-Box's pivotal role in introducing confusion during encryption [9][10][11][12].

In this context, a dynamic key generation method utilizing frame sequences is proposed. The receiving end employs the transmitting side's key to ascertain the frame order, initiating data decoding upon receiving data frames. The security of the key is bolstered with each additional data frame received. AES operates on data in 128-bit blocks (16 bytes) and supports three key sizes: 128, 192, and 256 bits. The number of rounds varies accordingly, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys,

and 14 rounds for 256-bit keys. Each round, except the final one, entails the execution of SubBytes, ShiftRows, AddRoundKey, and MixColumns operations, contributing to the overall security of the encryption process.

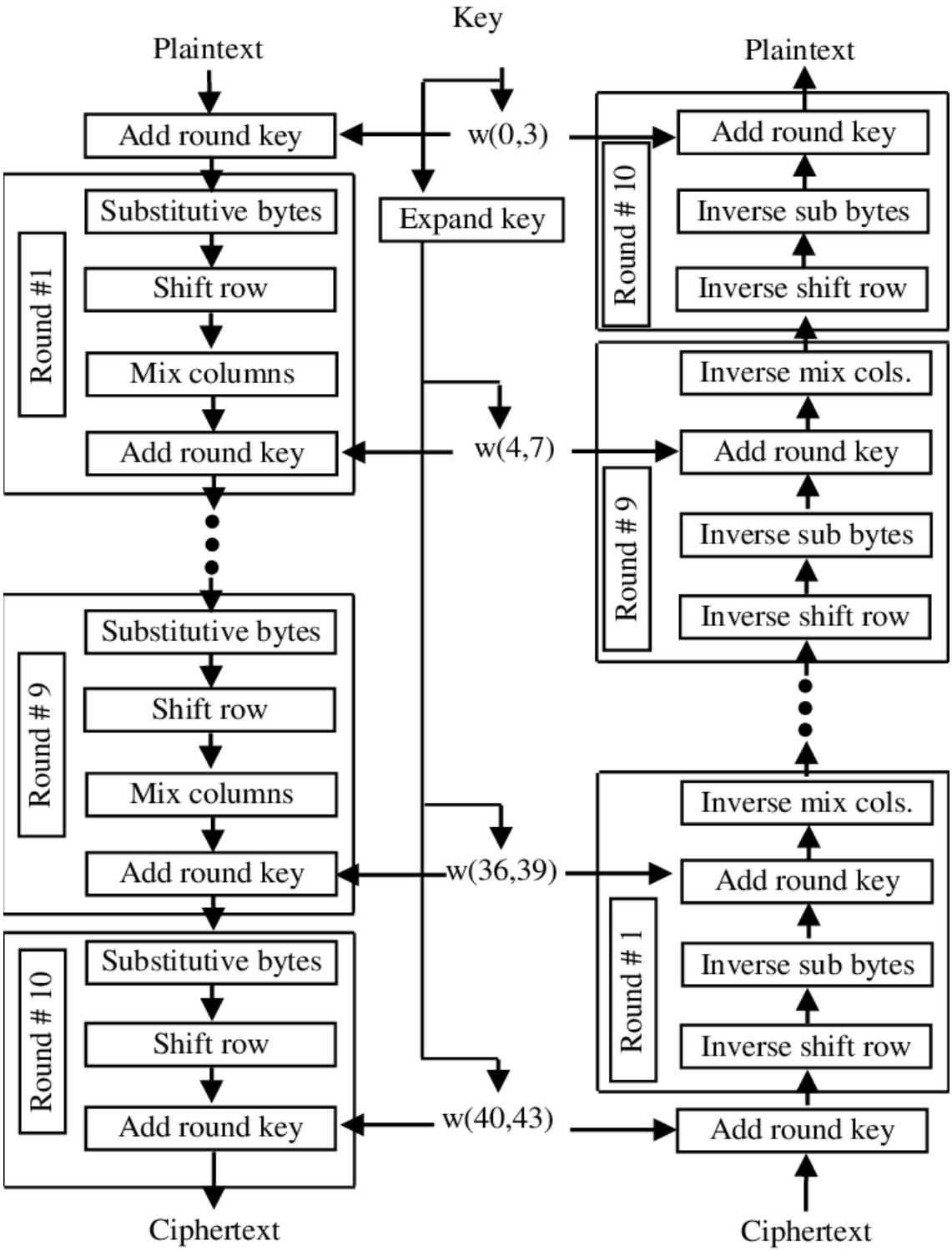

Figure 1: AES encryption and decryption

Related Work

The Advanced Encryption Standard (AES) has gained prominence due to its efficiency and security compared to encryption techniques such as IDES and 3DES [15]. AES, an internationally recognized symmetric block cipher encryption algorithm, has been widely adopted for its robustness [16]. In AES, a single matrix, known as a state array, is employed to process entire data blocks. Over time, researchers have introduced various enhancements to the original AES algorithm to bolster its security and data encryption capabilities [17]. These modifications encompass changes in key size [17], S-box design [18], and more.

Notably, efforts have focused on adapting AES to different hardware and software environments, as it has become the industry standard for encryption. Increasing the key size enhances resistance to cryptanalysis, and altering the S-box size reduces susceptibility to brute force attacks [18]. Numerous refinements have been made in earlier studies, significantly improving the algorithm's efficiency. However, ongoing research explores the potential of parallel implementation using Artificial Neural Networks (ANN) to encrypt vast volumes of data simultaneously, thus accelerating processing. Our work draws inspiration from a comprehensive review of prior AES modifications, leading us to propose a novel approach that incorporates dynamic keys to enhance data security.

In 2017, Usman et al. [18] introduced the Secure IoT encryption technique (SIT). This method employs a uniform substitution-permutation network with Feistel structure to create a symmetric block cipher. It utilizes a 64-bit key and can encrypt 64-bit plaintext in one operation. The 64-bit key undergoes permutation and f-function operations to generate the five keys required for the five rounds of this algorithm, each using a 16-bit key. The operations in each cycle involve swapping, bitwise XNOR with the key, f-function, and bitwise XOR. Implemented on inexpensive 8-bit microcontrollers, this technique offers a practical hardware solution.

In 2018, Chowdhury et al. [19] modified the AES method to enhance its portability and suitability for IoT contexts. Their approach focuses on utilizing a 1D S-Box generated using GF (24), as opposed to GF (28) used in the traditional AES S-Box. Experimental results showed an approximate 18.35 percent efficiency improvement over the original AES. This energy-efficient modification aligns well with the resource constraints often encountered in IoT environments.

Tsai et al. [20] introduced the Secure Low Power Communication (SeLPC) approach in 2018, aiming for low-power consumption with strong security. Their AES modification reduces the number of encryption cycles on end devices to conserve power. Additionally, it introduces a Dynamic Box (D-Box) update method instead of the traditional S-Box. The analysis demonstrates that SeLPC can withstand various attacks and reduce encryption power consumption by up to 26.2% compared to the original AES, making it suitable for IoT applications. In 2018, Hassan and Habeeb [21] proposed a secure IoT system for handling and monitoring medical information. Their approach secures data collected from IoT sensors using a modified version of the TSFS encryption method. Utilizing hardware platforms such as Arduino and Raspberry Pi, the system connects various sensors to protect the privacy and security of sensor data.

For IoT smart home applications, Khalaf and Mohammed [22] suggested two security services in 2018. The first ensures secrecy by encrypting all sensor data sent to the IoT server, employing AES-GCM or RSA-OAEP encryption techniques. The second service focuses on integrity, achieved through the SHA3-512 algorithm. Their assessment revealed that the AES-GCM method offers shorter average encryption times (around 3ms) compared to RSA-OAEP (around 9ms). Additionally, the integrity service consumes approximately 25% of the encryption time.

In 2019, Naif et al. [23] proposed lightweight modifications to the AES algorithm, utilizing a new chaotic system called JORN. Their approach replaces the MixColumns operation with multiple XOR stages, shift-cycle operations, and SHA3-256. MLAES, as it is termed, employs two 64-bit-sized S-boxes. K1 is responsible for shifting the S-boxes, providing fresh values at each cycle. Implementing MLAES using 40 sensors, the analysis revealed that it consumes less time and CPU cycles compared to the original AES.

These diverse modifications and approaches to AES encryption reflect the evolving landscape of IoT security, driving innovation and adaptability to suit the specific needs of various IoT applications.

Algorithm	Key Size (Bit)	Block Size (Bit)	Number of Rounds	Features
AES	128, 192 & 256	128	10, 12 & 14	Very high security and flexible
DES	64	64	16	Moderate security and flexible
3DES	112 & 118	64	48	High security and flexible
CALIFIA	128/192/256	128	18/22/26	High Security and Flexible
LED	64 & 128	64 &128	-	Efficient hardware implementation
PRESENT	80 & 128	128	32	Less gate count, less memory, Used for encrypting small amounts of data
RECTANGLE	80	64	25	Hardware friendly, faster and high throughput
TWINE	80 & 128	64	36	Ultra-lightweight and enough speed

TEA	128	64	32	Security can be enhanced just by increasing the number of iterations
HIGHT	128	64	32	Efficient hardware implementation and Ultra-lightweight
KATAN	80	32, 48 & 64	256	A hardware-oriented block cypher, inefficient software implementation, consumes too much energy and has low throughput.
Humming Bird	256	16	4	Suitable for RFID tags or Wireless Sensor networks, Low power consumption, High Speed
Blowfish	32 - 448	64	16	Very high security and flexible
Twofish	128, 192 & 256	128	16	It can't break remotely

Table 1: Comparison Lightweight Algorithms

Proposed Approach

In this section, we introduce our enhanced AES technique with dynamic keys, aimed at significantly improving security and establishing an active key management system. The primary objective of this proposal is to implement a compact and dependable symmetric encryption method to protect IoT sensor data.

Generation of Dynamic Keys

Addressing a common vulnerability shared by all symmetric key algorithms, including AES, necessitates a distinct key exchange approach. Therefore, we advocate the development of dynamic keys based on the order of data transfers. Both the sender and recipient must agree on the key before initiating data transmission. The process commences with data framing, where each frame comprises 16 bytes and is transmitted individually. To generate the dynamic key, we incrementally modify the 16-byte key in accordance with the sequence of the sent data frames. Specifically, each round involves XORing the 16-byte data frame with the respective sequence number before consecutive transmission and numerical indexing. This process within the proposed AES algorithm's dynamic key generation creates an activation key. For each data frame associated with a key change, the AES algorithm generates the Cipher Text, which is subsequently transmitted to the receiver. The recipient can deduce the key required for decryption based on the sequence in which the encrypted data is received, facilitating secure and synchronized data decoding.

The Encryption Method

In contrast to AES, the suggested encryption method uses round transformations such Divideswap, Subbytes, Shiftrows, Mixcolumns, and Xorroundkey. Introduced is a new round transformation called Divideswap, which splits the block into two halves and swaps the two halves. Five round transformations make up the suggested encryption method: I Divideswap; (ii) Subbytes; (iii) Shiftrows; (iv) Mixcolumns; and (v) Xorroundkey.

1. From the cypher key, create the set of round keys
2. Set the block data as the state array's initial value
3. The beginning state array should now include the initial round key
4. Carry out nine iterations of state modification

DivideSwap

SubBytes

ShiftRows

MixColumns

XorRoundKey

5. Tenth round of state manipulation should be carried out.

DivideSwap

SubBytes

ShiftRows

XorRoundKey

6. The encrypted data is copied out as the final state array (cypher text).

The following details each round's transformation:

1. DivideSwap: Composed of two transformations, such as splitting the block into two halves and exchanging the two halves, is the DivideSwap function.

2. SubBytes: Using the S-box table, a block's original byte is changed for a new one.

3. ShiftRows: The state array's rows can be changed with this function. The amount of bytes needed to rotate each row to the right is as follows:

• First Row: 0 byte rotation

• 2nd Row: 1 byte rotation

• 3rd row: 2 bytes of rotation

• 4th row: 3 bytes of rotation

4. MixColumns replaces each column of the state arrays C0 through C3 with a new column that was computed by matrix multiplication.

$$\begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \\ C'_3 \end{bmatrix} = \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 03 & 02 \end{bmatrix} \bullet \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

5. RoundKey XOR: State array XORs the value of the relevant round key, followed by an XOR with a sequence number, and replaces the state array with the outcome.

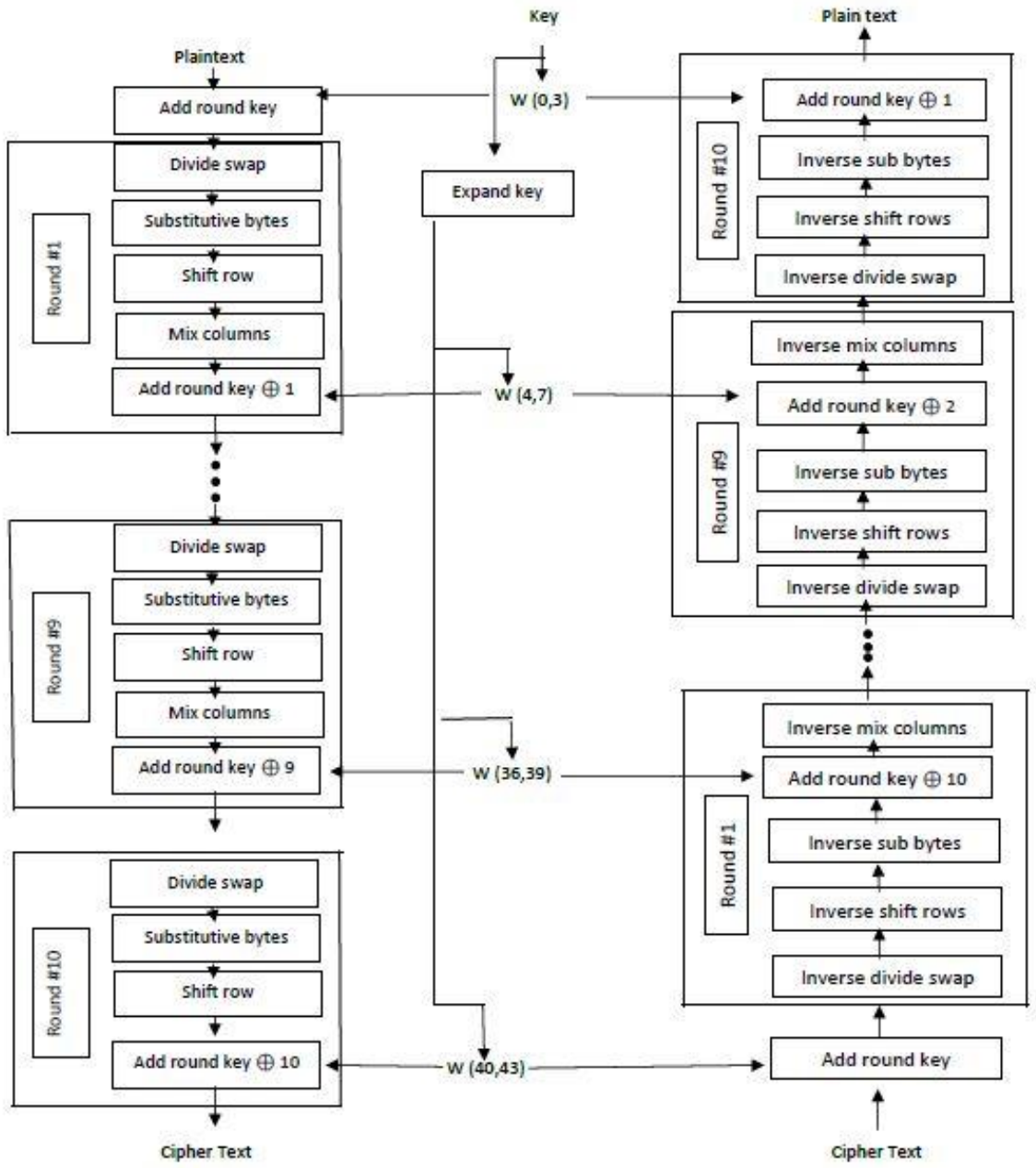

Figure 2: Proposed AES Encryption and Decryption

Decryption Scheme

Using inverse functions, decryption means undoing each stage of encryption:

Decryption is performed in the following order:

1. Execute the initial round of decryption:

InvDivideSwap

XorRoundKey

InvShiftRows

InvSubBytes

2. Complete nine complete decryption rounds:

InvDivideSwap

XorRoundKey

InvMixColumns

InvShiftRows

InvSubBytes

3. Execute the last XorRoundKey.

In the same sequence, the same round keys are utilised.

Implementation of Proposed Methodology

In this system, we have implemented a robust security mechanism using the Enhanced and Dynamic AES (ED-AES) encryption method. This encryption technique is instrumental in ensuring the protection of data gathered from the physical environment. To achieve this, we have incorporated specific hardware components, namely the Arduino Board, in conjunction with the ESP8266 module and DHT11 sensors.

The Arduino Board stands out as a user-friendly microcontroller device that possesses the capability to execute one application at a time while seamlessly connecting to a computer. What truly sets Arduino apart is its open-source nature, granting free access to a wealth of hardware and software development resources. This open-source approach fosters an environment of innovation and collaboration within the Arduino community, making it a cost-effective and versatile choice for our Internet of Things (IoT) security system.

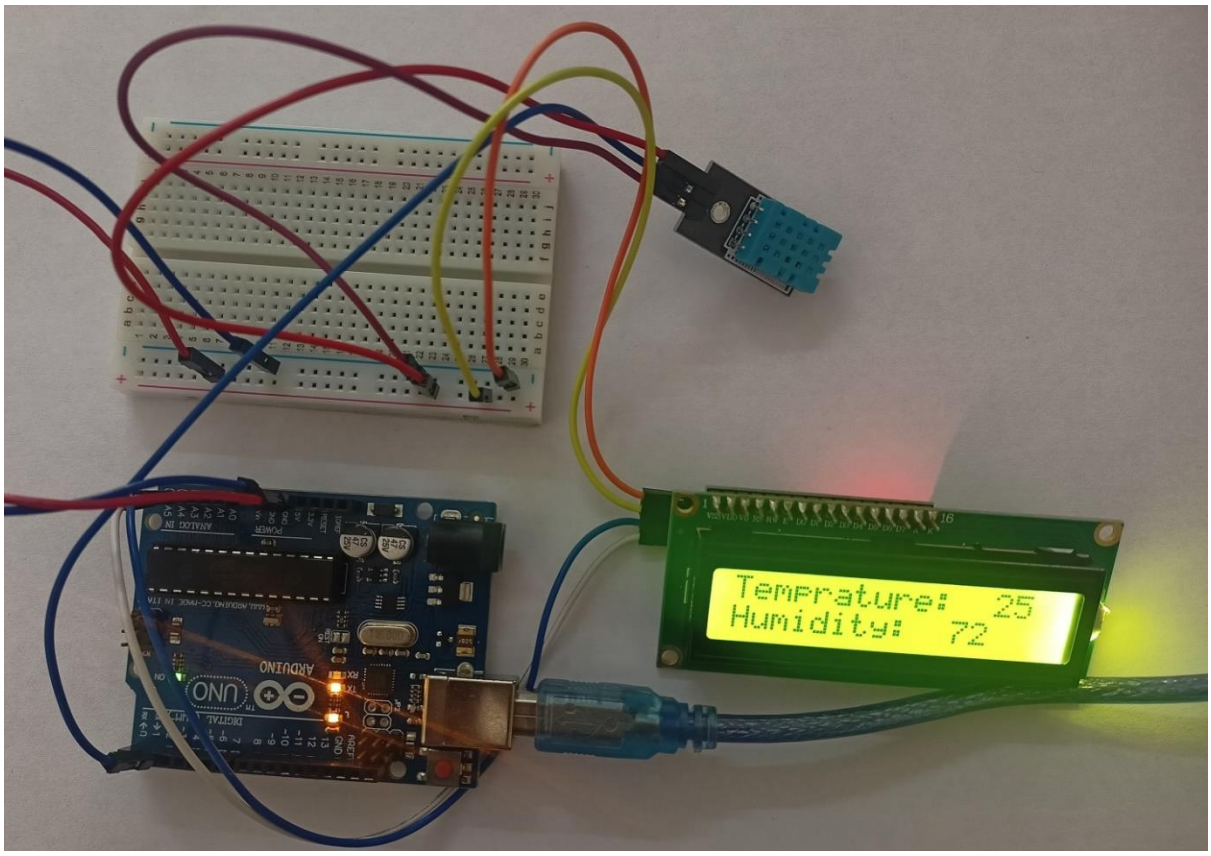

Figure 3: Read Temperature and Humidity using DHT11 and Arduino Uno

Depending on the requirements of the system, Arduino boards are adept at accepting both analog and digital input signals from a diverse array of sensors. These signals can then be translated into various outputs, including but not limited to initiating motor operations, toggling LEDs, establishing connections with cloud services, or performing a multitude of other functions. The Arduino UNO, undoubtedly the most renowned Arduino variant, is just one among several models available for selection [24, 25, 26].

To bolster the security of the sensing data obtained from the sensors connected to the Arduino, prior to its transmission across the network, we have integrated the ED-AES algorithm directly into the Arduino. This prudent measure ensures that sensitive data remains protected during transit, safeguarding the integrity and confidentiality of the information exchanged within our IoT ecosystem.

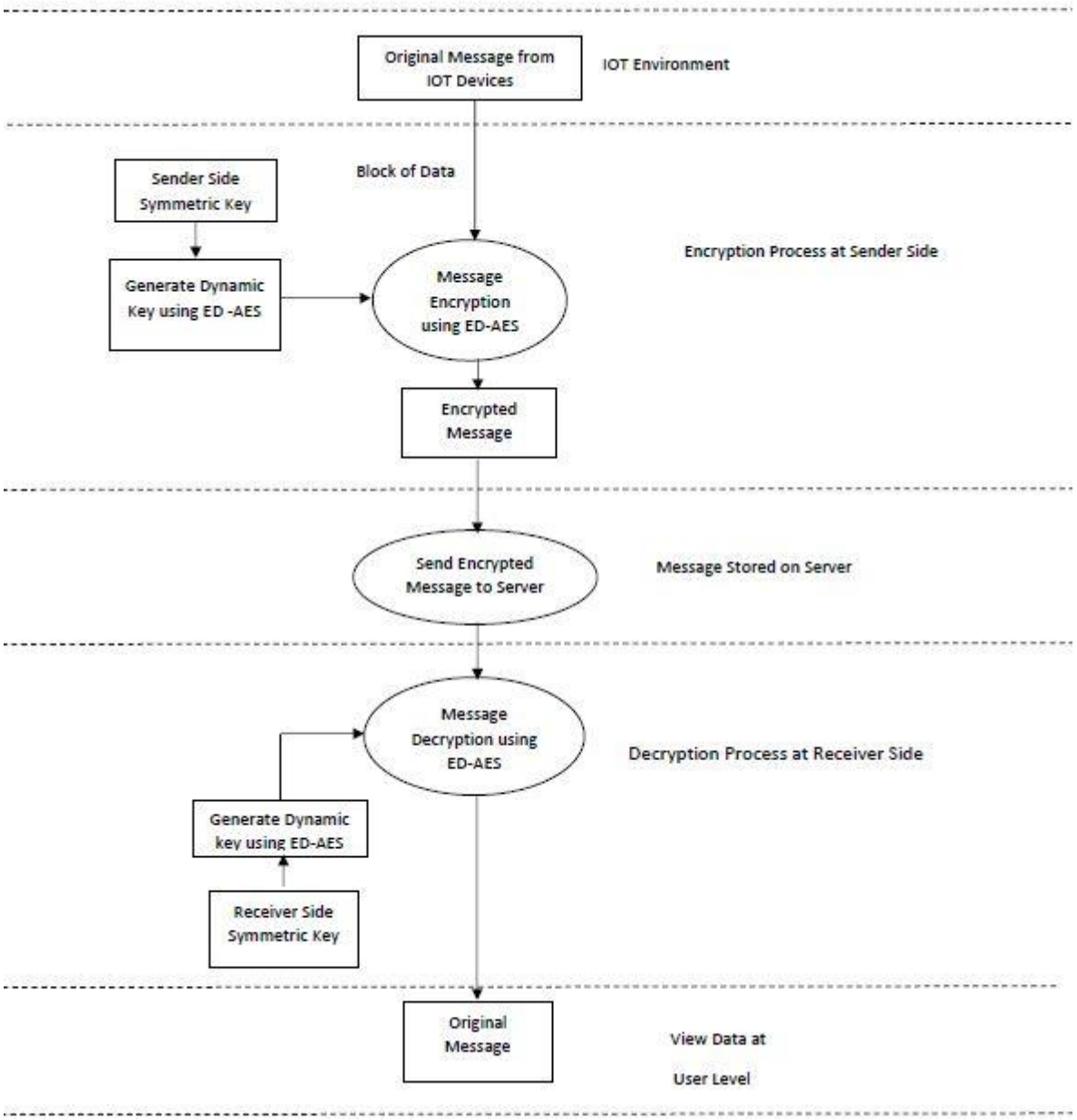

Figure 4: IoT Data Encryption and Decryption Process

This proposal is in charge of protecting sensor data transmitted over the network. Temperature and humidity are sensed by these sensors, which then use think-speak to relay the data to the server across the network. Temperature and humidity levels were measured using a DHT11 sensor. Data from these sensors will be sent to the server through the network.

Figure 5: Sending data from IoT Device to Server

The server's choices depend heavily on the data from this sensor. If an attacker has access to these data or has the ability to alter them, this results in different server choices and introduces mistake or wrongdoing into the system. As a result, the ED-AES algorithm included into the Arduino board will safeguard these sensor data.

Cryptanalytic Strength

The proposed ED-AES encryption method employs the same Galois field definitions as AES for its mathematical operations. However, it distinguishes itself from AES in a subtle yet significant manner, primarily through its round function. In ED-AES, the round functions utilize Divideswap and Roundkey XORed with the sequence number in each round. This distinctive approach enhances the algorithm's theoretical and statistical robustness. Consequently, it introduces greater complexity to both mathematical and statistical analyses.

With each round, this process alters the 256 possible options available to an element upon its introduction. Consequently, each choice made in the first round corresponds to a set of 256 alternatives. The same principle applies to subsequent rounds, meaning that for 10 rounds, there are a staggering 256^10 potential combinations of these options. This intricate design results in ED-AES encryption providing enhanced security compared to AES, owing to the additional layers of complexity embedded in the algorithm. Consequently, attempts to compromise the system's security through reduced-round attacks become less effective.

In each round of ED-AES, the application of Divideswap and Roundkey XORed with the sequence number fortifies the method's resistance against both differential and linear attacks. This meticulous integration of these elements contributes to the overall robustness of the encryption scheme, making it a formidable choice for safeguarding sensitive data.

4. Conclusion

We have modified the AES algorithm to enhance security and ensure its compatibility with a wide range of IoT devices and sensors. The proposed encryption technique incorporates various round transformations, including Divideswap, Subbytes, Shiftrows, Mixcolumns, and Xorroundkey. Furthermore, we have introduced the ED-AES algorithm, which employs a dynamic key generation technique. The data frame consists of sixteen sequential data elements, and their order holds significant implications. This key modification significantly enhances data transmission security for IoT systems.

To fortify the security of sensing data collected by Arduino-connected sensors before transmitting it across the network, we have integrated the ED-AES algorithm directly into the Arduino. This encryption method plays a crucial role in safeguarding the data obtained from the physical environment. The hardware components used for this proposal include the Arduino Board in conjunction with the ESP8266 and DHT11 sensors. Notably, the ED-AES encryption system has demonstrated equivalent cryptographic properties to AES while exhibiting robust resistance to potential cryptanalytic attacks. This complexity enhances the mathematical and statistical analysis involved in securing sensitive data.

References:

[1] S. Rajesh, V. Paul, V. G. Menon, and M. R.Khosravi, "A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices," symmetry mdpi, Vol. 11, No.2,2019.
[2] A. A. Ahmad, "A New Security Method for the Internet of Things Based on Ciphering and Deciphering Algorithms," Kirkuk University Journal/Scientific Studies(KUJSS), Vol. 13, No.3, 2018.
[3] K.L. Tsai, Y.L. Huang, F.Y. Leu, I. You, Y.L. Huang, and C.H. Tsai, "AES-128 Based Secure Low Power Communication for LoRaWAN IoT Environments," IEEE Access, Vol. 6, 2018.
[4] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," Springer -Verlag Berlin Heidelberg, 2017.
[5] J. R. Naif, GH A. Majeed, and A. K. Farhan, "Secure IoT System Based on Chaos- Modified Lightweight AES," International Conference on Advanced Science and Engineering (ICOASE), 2019.
[6] A. R. Chowdhury, J. Mahmud, A. R. M.Kamal, and M. A. Hamid, "MAES: Modified Advanced Encryption Standard for Resource Constraint Environments," IEEE, 2018.
[7] S. Habeeb and R. F. Hassan, "Build Secure Web of Things system to Manage Patient information Monitoring System," Iraqi Journal of Information Technology, Vol.9, No.1, 2018.
[8] S. B. Sadkhan and A. O. Salman, "Fuzzy Logic for Performance Analysis of AES and Lightweight AES," 2018 International Conference on Advanced Science and Engineering (ICOASE), Duhok, 2018, pp. 318-323, DOI: 10.1109/ICOASE.2018.8548832.
[9] A. K. Farhan, R.S. Ali, H. Natiq, and N. M.G. Al-Saidi, "a new S-Box generation algorithm based on Multistability behaviour of a plasma perturbation model," IEEE Acess, Vol. 7, 2019.
[10] A. H. Zahid and M. J. Arshad, "An Innovation Design of Substitution- Boxes using Cubic polynomial mapping," mdpi symmetry, Vol. 11, No.3, 2019.
[11] A. H. Zahid, M. J. Arshad, and M. Ahmad, "A Novel Construction of Efficient Substitution-Boxes Using Cubic Fractional Transformation," mdpi entropy, Vol.21, No. 3, 2019.
[12] A. Alasaad and A. Alghafis, "key-dependent S-box Scheme for Enhancing the Security of Block Ciphers," 2019 2nd International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 2019, pp. 1-4, DOI: 10.1109/ICSPIS48135.2019.9045900
[13] N. Mathur and R. Bansode, "AES Based Text Encryption Using 12 Rounds with Dynamic Key Selection," Elsevier, 7th International Conference on Communication, Computing and Virtualization, Procedia Computer Science, Vol.79,2016.
[14] A. M. Raheema, S. B. Sadkhan, S. M. Abdul Sattar, “Performance Evaluation of Voice Encryption Techniques Based on Modified Chaotic Systems”, 2020 6th International Engineering Conference “Sustainable Technology and Development"(IEC), PP: 135-140
[15]. Douglas Select"Advance Encryption Standard” Insight: rivier academic journal, volume 6, number 2, fall 2010.
[16]. Priyadarshini Patil, Prashant Narayankar, Narayan D G, Meena S M” A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish” International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA.
[17]. Ahmed Tariq Sadiq, Faisal Hadi Faisal" Modification AES algorithm Based on Extended Key and Plain Text” Design for scientific renaissance ISSN: 2231-8852
[18]. M. Usman, I. Ahmed, M.I. Aslam, S. Khan, and U. A. Shah 2017.‖ SIT: A Lightweight Encryption Algorithm for Secure Internet of Things, ‖International Journal of Advanced Computer Science and Applications (IJACSA), 8(1).
[19]. A. R. Chowdhury, J. Mahmud, A. R. M. Kamal, and M. A. Hamid 2018. ―MAES: Modified Advanced Encryption Standard for Resource Constraint Environments,‖IEEE.
[20]. K.L. Tsai, Y.L. Huang, F.Y. Leu, I. You, Y.L. Huang, and C.H. Tsai 2018. ―AES-128 Based Secure Low Power Communication for LoRaWAN IoT Environments,‖IEEE Access, Vol. 6.
[21]. S. Habeeb and R. F. Hassan 2018. ―Build Secure Web of Things system to Manage Patient information Monitoring System,‖ Iraqi Journal of Information Technology, 9(1).
[22]. R. H. Khalaf and A. H. Mohammed 2018. ―Confidentiality and Integrity of Sensing Data Transmission in IoT Application,‖ International Journal of Engineering & Technology, 7: 240-245.
[23]. J. R. Naif, G.H. A. Majeed, and A. K. Farhan 2019.― Secure IoT System Based on Chaos- Modified Lightweight AES, ‖ International Conference on Advanced Science and Engineering (ICOASE), 2019.
[24]. M. Usman, I. Ahmed, M.I. Aslam, S. Khan, and U. A. Shah 2017.‖ SIT: A Lightweight Encryption Algorithm for Secure Internet of Things, ‖International Journal of Advanced Computer Science and Applications (IJACSA), 8(1).
[25]. R. H. Khalaf 2019. ―Secure Mechanisms for Smart Home IoT Application,‖ Thesis.
[26]. Arduino Tutorials point 2016. ‖ ©Copyright 2016 by Tutorials Point (I) Pvt. Ltd.