IoT Security Using AES Encryption Technology based ESP32 Platform

Mohammad Al-Mashhadani
Department of Computer Engineering,
Middle Technical University (MTU), Iraq
bbc0034@mtu.edu.iq

Mohamed Shujaa
Department of Computer Engineering,
Middle Technical University (MTU), Iraq
drshujaa@mtu.edu.iq

Abstract: *The Internet of Things (IoT) is one of the most important modern technologies that have attracted the most interesting areas of life, whether industrial, academic, or other, in recent years. The main goal is to integrate the physical world with the digital world through a seamless ecosystem, and this constitutes a new era for the Internet. This technology provides high commercial value to enterprises as it provides many opportunities in many applications such as energy, health, and other sectors. However, this technology suffers from many security problems, as it is considered the biggest challenge due to its complex environment and the limited resources of its devices. There is a lot of research to find successful security solutions in IoT, in this research, a proposed solution to secure IoT systems using Advanced Encryption Standard (AES) technology is achieved. Some sensors were linked as an example of the Internet of Things. The data is received by the card created and developed by Espressif Systems (ESP32) module, where its encrypted then sends to the internet site through an authorized person to be received from anywhere, and it is also possible to receive it via a published IP which is announced within the internal network of the ESP32 device module. The decryption part is proposed at last to find out the true values of the sensors. The proposed approach shows good secured and balanced results at the end.*

Keywords: *ESP32, IoT Security, secure boot, AES.*

Received May 27, 2021; accepted September 9, 2021
https://doi.org/10.34028/iajit/19/2/8

1. Introduction

Internet of Things (IoT), works to provide various services by linking people on the one hand with sensors IT components and actuators on the other hand, these services are widely used. Also, the number of connected devices in the Internet of Things is increasing tremendously. Whereas in 2015, nearly 15 billion devices were connected, and in 2019 they reached nearly 26 billion, and the number may reach about 75 billion devices by 2025, and the Internet of things market around the world has doubled since 2016, and forecasts estimate. Whereby 2020 it could reach about $ 457 billion [7].

With the exponential growth of IoT, there is a growing problem of security threats. There are some reasons that make devices connected to the Internet of things vulnerable to these threats and attacks, including:

- An attacker can have physical access to these devices, as most of them operate without human intervention.
- Attackers can eavesdrop on these devices because they are linked to wireless networks among themselves.
- These devices do not support complex security algorithms due to the nature of their installation, which depends on a low power level, as well as for the low computational capabilities [21].

IoT technology can be one of three types:

- Directed via the Internet which, means that it works as an intermediary program.
- Object-oriented means that it provides the ability of sense.
- Semantic vector and this type allow access to knowledge, and this type depends on the working principles of a specific application. A mixture of types or just the independent Internet of Things can be used to build smart applications aimed at solving problems in everyday life [4].

To ensure IoT security, the following objectives must be met:

- Confidentiality: (protecting data from unauthorized disclosure).
- Safety: (ensuring cannot be modified that data if not obtained prior permission).
- Availability: (ensure access to data as needed).

To ensure confidentiality, encryption algorithms are used. To ensure safety, Message Authentication Code (MAC) devices and digital signatures are used. Availability cannot be provided through encryption techniques.

There are many areas of cybersecurity in the Internet of Things. Cryptography is an important technology used to secure data and transactions in the

Internet of Things.

For encryption, themost commonly used algorithms are Advanced Encryption Standard (AES) for symmetric encryption, Rivest Shamir Adleman (RSA) for asymmetric encryption, and for a digital signature algorithm, the Digital Signature Algorithm (DSA) or ECDSA (elliptic curve DSA) is used [7].

The AES: is one of the methods used to achieve privacy and confidentiality of data that is transferred over various computer networks.

The AES algorithm is one of the most widely used and symmetric block cipher algorithms used worldwide. This algorithm includes a special structure for encryption and decryption of sensitive data. It is possible to implement AES in software or hardware. This algorithm is implemented on devices to meet the speed, security and throughput requirements of communication systems used in modern applications. Moreover, it is very difficult for hackers to get the original data when encrypting with this algorithm. So far, there is no evidence that this algorithm is broken. The AES algorithm has the ability to handle three different key sizes (128, 192, and 256) bits [2, 12].

AES-128 encryption is implemented on the ESP32 card, as this chip has many advantages as it includes Wi-Fi and Bluetooth as well as a 32-bit CPU, a number of General Purpose Input/Output (GPIOs), and support for a variety of protocols such as SPI (Serial Peripheral Interface), Inter-Integrated Circuit Protocol (I2C), Universal Asynchronous Receiver-Transmitter (UART), and it is designed to support the work of projects Internet of things.

1.1. Internet of Things Solutions and Applications

1.1.1. Classification of Security Solutions in the Internet of Things

Since the issue of security is one of the most important problems that researchers focus on in the topic of the Internet of things, there are many proposals aimed at addressing these problems. Solutions can be classified from an architectural point of view in the Internet of Things, as shown in Figure 1, into two main classes of solutions:

- Classical Approaches: Includes solutions of cryptographic based technologies specifically designed for IoT connections or adapted from wireless sensor networks or Machine to Machine (M2M) connections. It is worth noting that most of these solutions work in a centralized environment, meaning that we have reliable central entities that can ensure the proper performance of security services. Cipher tools are divided into symmetric or asymmetric techniques.
- Emerging new security solutions: This category includes solutions based on new technologies other than coding techniques. These solutions are better suited to face scalability problems as compared to encryption methods, as these solutions operate in a decentralized environment [15].

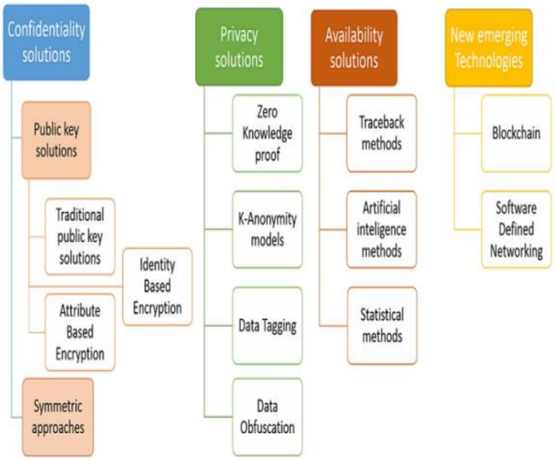


Figure 1. Two main classes of solutions.

1.1.2. Applications and Challenges in the Internet of Things

Many applications have improved due to the Internet of things in various fields, such as healthcare, smart cities, smart homes, smart networks, in addition to other industrial applications. However, the introduction of IoT devices as well as restricted IoT technologies in such sensitive applications leads to new privacy and many security challenges. Figures 2 and 3 illustrate some of the applications and challenges [15].

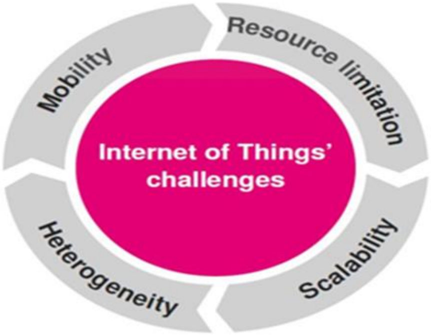


Figure 2. IoT challenges.

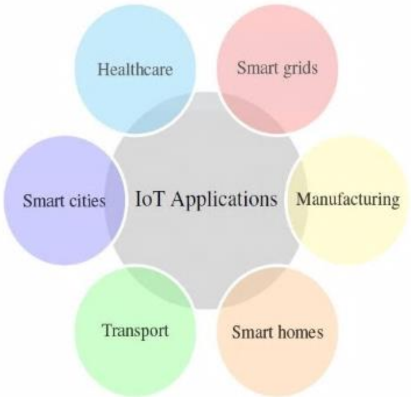


Figure 3. IoT application.

1.2. Advanced Encryption Standard (AES)

1.2.1. Why AES Algorithm?

- AES algorithm is trusted as a standard by the US government as well as many institutions.
- AES is the most widely used, and most popular today.
- Compared to Triple Data Encryption Standard (TripleDES), it is more than six times faster.
- In terms of cyber security, AES is the most accepted encryption standard in the world.
- It is used in messaging applications such as Signal and Whatsapp, computer platforms such as VeraCrypt, and other commonly used technologies.
- One key is used for encryption and decryption so both sender and receiver have the same key.
- The key sizes are suitable for securing classified data to a satisfactory confidentiality level.
- Although it is highly efficient in 128-bit format, it also uses (192 and 256) bit keys to encrypt heavy-duty tasks.
- AES can largely be considered impervious to all attacks, except for brute force, which attempts to decrypt messages using all possible combinations of 128, 192 or 256-bit encryption [19].

1.2.2. Evaluation Criteria

The criteria used by National Institute of Standards and Technology (NIST) are three important criteria, sent by experts to evaluate algorithms, that were the reason for introducing the AES algorithm, these criteria are:

- Security: This algorithm has a high ability in the matter of data protection, as the main objective of the algorithm was to improve the security problem in the DES algorithm.
- The cost: Also, one of the main goals of this algorithm was to improve the low performance of the DES algorithm Therefore, it has high computational efficiency and uses a very wide range of applications.
- Implementation characteristics: NIST has evaluated some important aspects of the algorithms, namely the simplicity, flexibility, and suitability of the algorithm to the diversity of applications [2].

2. Problem Statement

Most of the problems and challenges with the Internet of Things are privacy and security. Due to its extreme complexity and the very large number of beneficiaries of this technology, as well as the increase in specific applications, in addition to the fact that the technologies that work with it are constantly changing. Therefore, it is necessary to find successful and appropriate solutions for the development of this technology and at the same time contribute to increasing security and privacy.

3. Related Work

- Kodali and Soratkal [14], proposed a mechanism linking the MQTT (Message Queuing Telemetry Transport) to the ESP8266 chip, and they also relied on open source code in this system, for remote monitoring in the smart home application. Also, no safety and protection technology was applied, and the proposed system was run on the computer to reduce the security risks.
- Nandhini and Vanitha [16]. He proposed a new, lightweight and compact hybrid coding technology through the use of Faster Bit Switching instructions with the S-box of PRESENT.
- Abd Zaid and Hassan [1]. They proposed a modified algorithm for AES with a lightweight problem, the mixed column process was combined in this proposal with the process of adding around key in one cycle, and they also modified the shift row process to shift rows and shift columns, and the result was that the number of rounds was reduced to six Only, this proposed algorithm passed statistical tests at a higher speed than the AES standard, and it is also applicable to IoT resources.
- Parida *et al.* [17]. They proposed designing a system that monitors (air quality, temperature, and humidity) in real-time using the Internet of Things (IoT), and the data received is stored on the Thing Speak cloud. This data can then be utilized and analyzed in different applications, and the ESP8266 chip was used to transfer data from sensors to the cloud. In this paper, it is noted that there is no focus on the security aspect of the Internet of Things, in addition, that the use of ESP32 is better and more efficient than ESP8266.
- Hussein and Shuja [11] they suggest encoding and decoding secure messages between nodes via MQTT protocols using a one-time board as well as DNA (Deoxyribose Nucleic Acid) computing techniques. The messages produced by the sending node are encrypted and the message is decoded and rearranged by the receiving node.
- Khoa *et al.* [13], they proposed a mechanism to improve IoT security efficiency by using the SHA-256 (Secure Hash Algorithm 256-bit). This mechanism is represented by encrypting the user name, along with a password and a code, to authenticate a specific device with a web server, and a monitoring system has been put in place that allows remote control of IoT devices.

4. Proposed System

A group of sensors were connected to represent the IoT part, and these sensors (DH11 temperature andhumidity sensor, ultrasound sensor, and LED lights) were linked to the ESP32, the ESP32 chip receives and encrypts the data, and then this data is sent to the special internet page , where this page is opened securely by a name and a password, the data is also published on the IP address within the local Internet network linked to the ESP32 chip within the same service provider, after receiving the data it can be decrypted byusing the AES application on the Internet and also as an application on a mobile devices, to know the real values of the data received, and in this way it will be difficult to know the real values in the event that someone can access the information page to steal the data, because he will need to know the algorithm used for encryption and also the key used in this method, and in this way security can be strongly achieved and the information is protected, Figure 4 show this design.

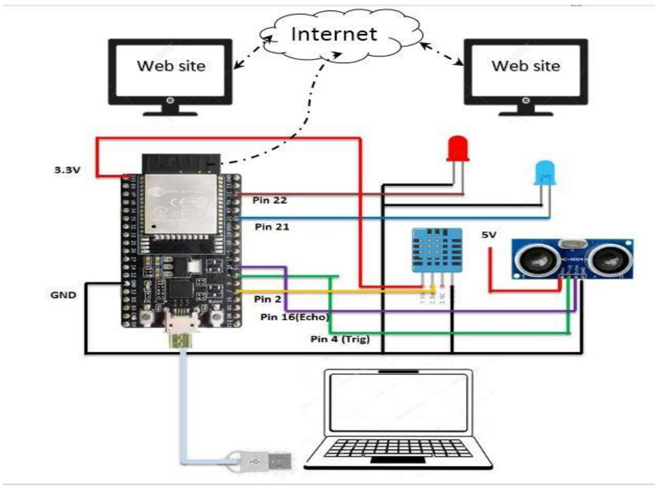


Figure 4. Proposed design.

4.1. ESP32 Module

4.1.1. Overview

ESP32: A combined 2.4GHz Wi-Fi and Bluetooth chip, built with 40nm ultra-low-power TSMCtechnology. This is to achieve the best performance in addition to the Radio Frequency (RF) performance, and is characterized by reliability, durability, and versatility in a variety of power scenarios and applications [6]. ESP32 series chips include ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD, ESP32-U4WDH, including ESP32-D0WD-V3 and ESP32-D0WDQ6 based The ESP32-U4WDH is on the ECO V3 chip, See Figure 5 [5].

Figure 5. Esp32 WROOM-32S.

4.1.2. ESP32 SoC (Block Diagram)

The ESP32 SoC shown in the figure contains:

- 12 Xtensa 32-bit cores (called PRO and APP CPU) at 240MHz RTC13 subsystem with ULP.
- 530 KB SRAM and 448 KB ROM.
- FAST RTC SRAM (8 KB) and SLOW RTC SRAM (8 KB).
- eFuse memory (1 KB).
- Built-in radios: Bluetooth Low power radios, Bluetooth 4.2, and Wi-Fi 802.11 / b / g / e / I.
- Surround input/output: UART, SPI, Ethernet, I2C, ADCs, DAC, capacitive Touch sensors, PWM (Pulse-width modulation), etc.
- Contains that optional Flash Two built-in chip variants included Flash-ESP32-PICO-D4 with (4MB) and ESP32-D2WD with (2MB).
- HW Encryption Accelerator with AES,SHA-256,RSA, and RNG. See Figure 6 [22].

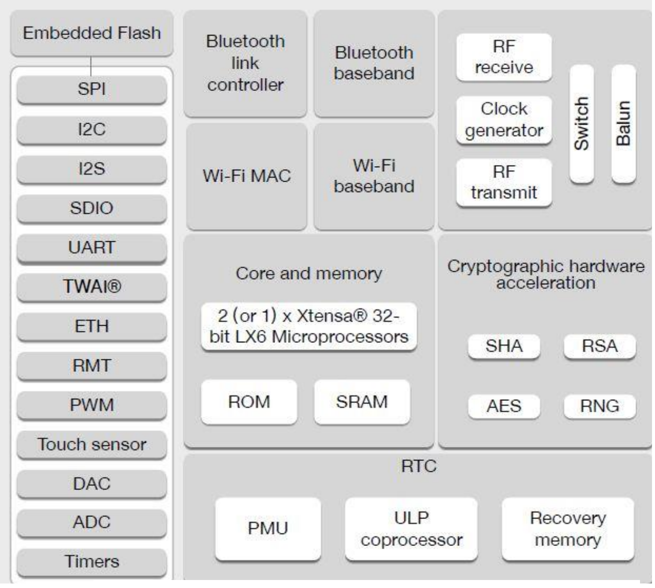


Figure 6. Esp32 SoC (Block Diagram).

4.1.3. Pin Description

There are 39 pins on the ESP32, 34 of which are used as GPIO and the rest are for input only. This ESP32 chip supports 18 channels for 12-bit ADC as well as 2 channels for 8-bit DAC. The IT has 16 channels that generate a PWM signal and 10 GPIO pins support the capacitive touch feature. Also, ESP32 has a multiplexing feature, through this feature the programmer can configure any GPIO pin for serial communication or PWM by programming it. (ESP32) supports (2) I2C interfaces, (3) SPI interfaces, and (3) UART interfaces, and supports CAN (Controller Area Network Protocol) [12].

a) UART interface: esp32 contains three pairs of Rx and Tx pins that support UART interfaces for TTL (Transistor–transistor logic) connections. These pins can be dealt with by software, and any GPIO pin can be used as aUART by programming it.

b) External interrupt: Since the ESP32 has a multiplexing feature, any GPIO pin can be programmed if we want to use it as an interrupt pin.
c) GPIO23(MOSI), GPIO19(MISO), GPIO18(CLK), GPIO5(CS): Can be used for SPI communication. Since the ESP32 has (2) setsof SPI's this is the first
d) GPIO13(MOSI), GPIO12(MISO), GPIO14(CLK), GPIO15 (CS): Can be used for SPI communication. This is the second group that the ESP32 from SPI.
e) GPIO21 (SDA- Serial Data Line), GPIO22 (SCL- Serial Clock Line-): ESP32 uses these two pins for IIC connections through the Wire library.
f) Reset Pin: Used for resetting in the ESP32 chip is the enable pin When LOW it will reset the console.

These posts and pins are shown in Figure 7.

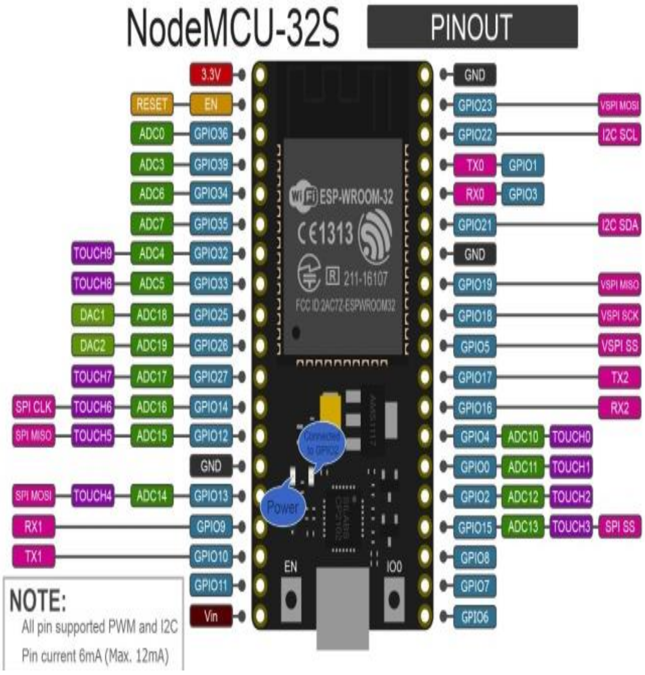


Figure 7. Pins descriptions.

4.1.4. ESP32 VS ESP8266

The ESP32 chip has some advantages that make it superior to the previous version (ESP8266) as shown in the Table (1), [18]:

Table 1. Comparison between Esp 32 and Esp 8266.

No.	Attribute	Esp 8266	Esp 32
1	Voltage/Current consumption	3.3V/ 10uA ~ 170mA	3.3V / 10uA ~ 260mA
2	Processor/ Processor speed	Xtensa® Single-Core 32-bit L106/ 80-160MHz	Xtensa® Dual-Core 32-bit LX6 600 DMIPS / Dual 160MHz
3	GPIO	17	36
4	SRAM	160kB	512kB
5	Hardware / Software PWM	0/8 channels	1/16 channels
6	A / D converter	1x10bit	7x12bit
7	Max TCP	5	16
8	Support 802.11	b / g / n / d / e / i / k / r	11b / g / n / e / i
9	SPI / I2C / I2S / UART / CAN	2/1/2/2/0	4/2/2/3/0
10	Bluetooth	---	Bluetooth 4.2
11	MAC interface	---	1

4.2. IoT Sensors

4.2.1. DHT11-Humidity and Temperature Sensor Specification

a) The operating voltage is: 3.5 to 5.5 V.
b) The operating current is: 0.3 mA (measuring) and 60 µA (standby).
c) O/P: serial data.
d) Temperature Range : 0ºC to 50ºC.
e) Humidity Range: from 20% to 90%.
f) Resolution: (Humidity and Temperature)16-bit.
g) Accuracy: (±1ºC) and (±1%).

Figure 8 shows the shape of the sensor, and Table (2) shows the Pin Configuration [9].

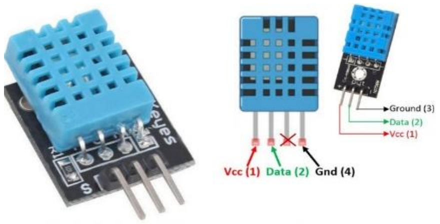


Figure 8. DHT11 sensor.

Table 2. Pin Configuration and identification.

No:	Pin Name	Description
DHT11 Sensor		
1	VCC	Power supply 3.5v to 5.5 v
2	Data	Outputs both Humidity and Temperature through serial Data
3	NC	No Connection.
4	Ground	It connects with the ground in the circuit
DHT11 Sensor module		
1	VCC	Power supply 3.5v to 5.5 v
2	Data	Outputs both Humidity and Temperature through serial Data
3	Ground	It connects with the ground in the circuit

4.2.2. HC-SR04 Ultrasonic Sensor specification

a) The operating voltage is: +5v.
b) Theoretical measuring to distance: from (2cm) to (450 cm).
c) Practical measuring to distance: from 2cm to 80cm.
d) Accuracy: 3mm.
e) Measuring angle covered: <15º.
f) The operating current is: <15mA.
g) The operating frequency is:40 Hz.

Figure 9 shows the shape of the sensor and Table (3) shows the Pin Configuration [10].

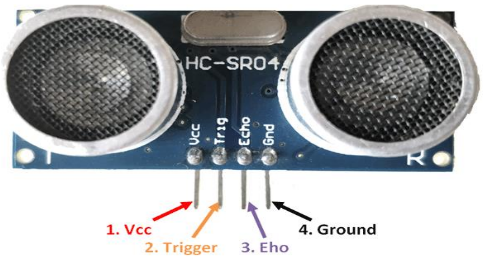


Figure 9. HC-SRO4 ultrasonic sensor.

Table 3. Ultrasonic sensor pin configuration.

No	Pin Name	Description
1	VCC	Power supply + 5 v
2	trigger	A trigger pin is an i/p pin that has to be kept high for 10µs for initialize measurement by sending as a wave.
3	Echo	The echo pin is an o/p pin. This pin also goes high for some of the time that will be the same as the time taken for the wave to return back to sensor.
4	Ground	It connects with the ground in the circuit

4.3. Advanced Encryption Standard (AES) Cryptography

4.3.1. (AES) Specifications

AES Algorithm: It is a symmetric key block cipher technology used for encryption/decryption. This technology uses 128, 192, and 256-bit encryption keys to encrypt/decrypt 128-bit data blocks. In this paper, we used a 128-bit key length for encryption/decryption. Using a 128-bit encryption key in this algorithm provides high security because an encryption key of this length is difficult to crack. As the estimated time required to break this key is years. The AES algorithm consists of the original key, metadata, and ten rounds (128 bits). The user gives the original key and raw data randomly. In this algorithm there are nine rounds, four operations are performed on the data in each of these nine rounds in the two states of encoding and decoding.

In the case of encryption, the operations that are performed are: byte substitution (S-box), ShiftRows, MixColumns, and add round key. In the tenth (last) round, the same operations are performed except for the MixColumns transformation.

In the case of decryption, the previous operations are performed in reverse, the reverse operations are more complex compared to the same operations in the encryption. The number of rounds in this algorithm depends on the length of the key [23].

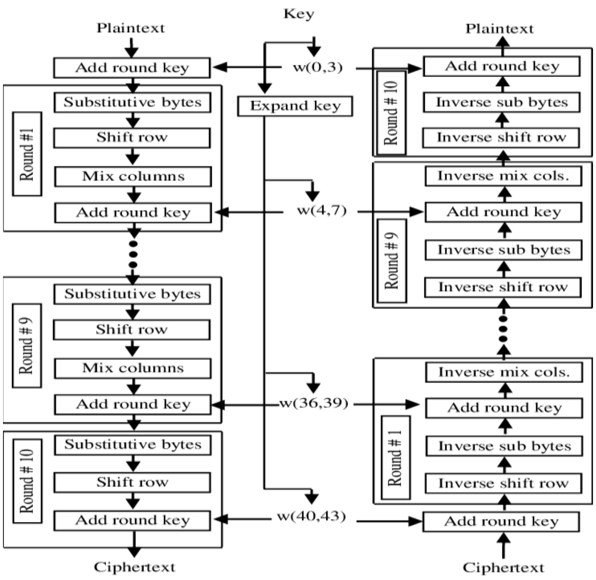


Figure 10. Description of the AES algorithm.

4.3.2. Description of the AES Algorithm

1. Key Expansions- Round keys are derived from the cipher key using the Rijndael key table. AES requires a separate 128-bit round key block for each round plus one more. Figure (10) illustrates how this algorithm work
2. Round Primary: Add Round Key-Each byte of the state is combined with a round key block using bitwise xor.
3. Both the encryption and decryption processes require a number of rounds, the number of rounds depends on the length of the key used in encryption and decryption, in the research paper we used a key length (128 bits) so this algorithm needs 10 rounds to perform the encryption or decryption process, and in each round of them Four operations are performed, except for the last round in which three operations are performed as described below.

   a) Rounds: (1-Sub Bytes, 2-Shift Rows,3-Mix Columns, 4-Add Round Key).
   b) Final Round (1-Sub Bytes, 2-Shift Rows, Add Round Key) [23].

As for the decryption process, the operations that are executed in each round are:

a) Rounds: (1-InvSubBytes, 2-InvShift Rows, 3-InvMix Column, 4-Add Round Key).
b) Final Round: (1-InvSub Bytes, 2-InvShift Rows, 3-Add Round Key)

4.3.3. Substitute

(Sub Bytes and InvSub Bytes) are the first operation in each run of execution, in which each byte of the state is replaced by a byte of nonlinear S-box and Inverse-S-box, to find the replacement value, the intercept must be used in the table. Sub Bytes and InvSub Bytes are used to hide the relationship between plain text and ciphertext [2, 3] see in Figure (11).

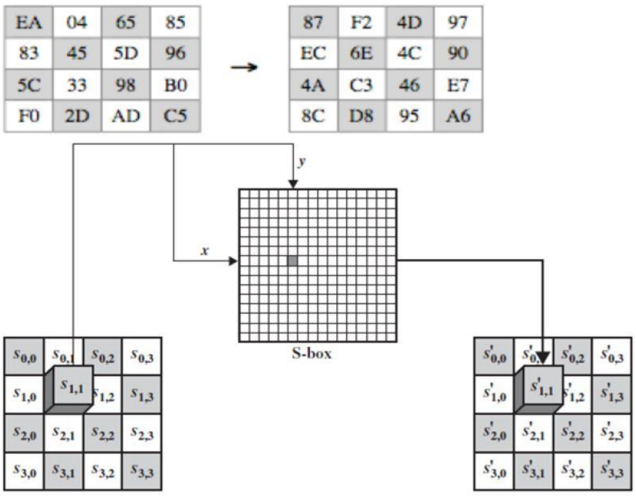

EA 04 65 85
83 45 5D 96
5C 33 98 B0
F0 2D AD C5
87 F2 4D 97
EC 6E 4C 90
4A C3 46 E7
8C D8 95 A6
y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Figure 11. Substitute byte transformation.

4.3.3.1. Shift Rows

In this process, the state bytes are periodically shifted to the left for every row except for the first row of the state array. One byte in the second row is shifted circularly to the left. 2 bytes are shifted to the left in the third row. In the fourth (last) row, the shift is three bytes to the left [20]. In this process, the size of the new state is not changed and remains the same as the original 16-byte size, but the change is in the location of the bytes, As for the InvShiftRows operation during decryption, it is done in the same way but in the opposite direction. as in Figure (12) [8].

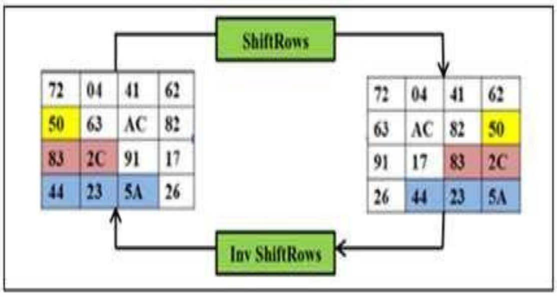


Figure 12. Shift Row for encryption and decryption.

4.3.3.2. MixColumn

In (MixColumns and InvMixColumns) the state matrix column is multiplied by a constant matrix C (x), as in Figure (13), the value of the matrix (B), in both the encoding and decoding process. is the product of [A] and a constant [C (x)], the only difference between the two operations is the value of [C(x)], as in equation (1) for encryption, and (2) for decryption [8].

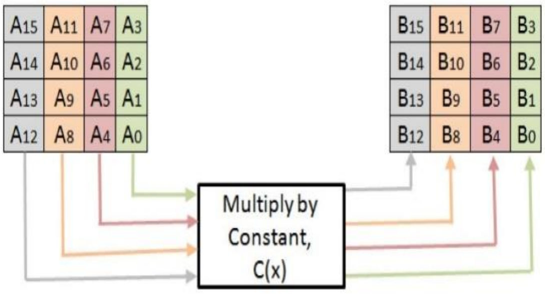


Figure 13. MixColumn multiplied by C(x).

4.3.3.3. Add Round Key

Works in both encryption and decryption in the same way. 128 bits are added from the generated key, as in Figure (14) [8].

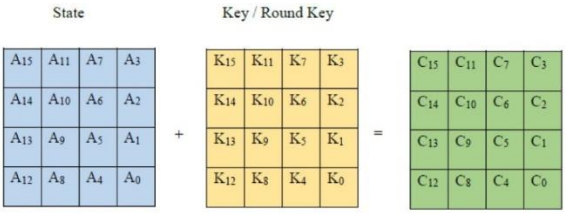


Figure 14. Add Round Key operation adds 128-bit.

5. Result and Discussion

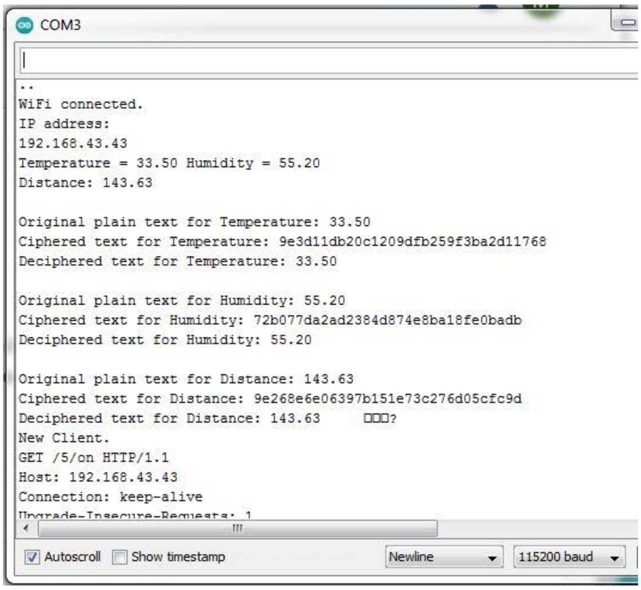
Praise be to God, the program was implemented on the design shown in Figure 4, noting that only the encoded values or the original and encrypted values can be sent. The data was received from the sensors via the ESP32 chip, where the AES 128bit algorithm was used to encrypt the data and display it on the Serial Monitor in the Arduino program, and the data was sent to the cloud on Thing Speak and Firebase, in addition to publishing it on a local IP address through which Web screen display including data values with the ability to control 2 LEDs.

The results we obtained can be compared with some related previous works as follows:

For example in [14] the ESP8266 and MQTT for remote monitoring are integrated into a smart home. However, no IoT security mechanism has been implemented. In addition, ESP8266 is less efficient than ESP32. In [17] the data from sensors is received by ESP8266 and stored in the cloud, without using IoT security mechanism. In addition, ESP8266 is less efficient than ESP32. In [13] a remote home control system was designed and implemented, as an example of the Internet of Things, where a set of sensors are connected, and information is received through ESP8266, in addition to using a security mechanism (SHA256).

As for the results we obtained in this paper, they can be presented as follows:

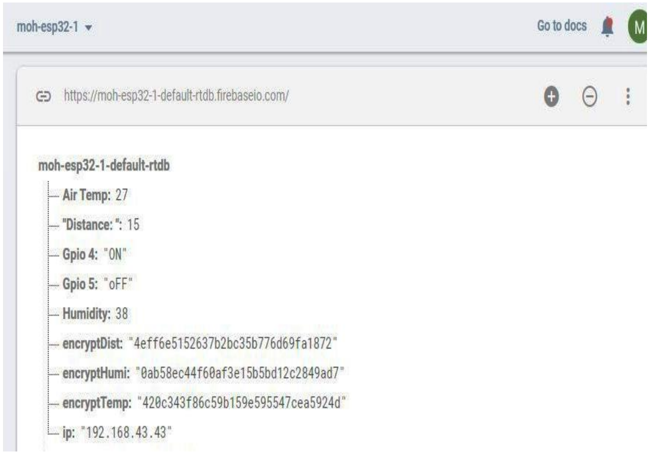
- Figure 15: representing the serial interface of the Arduino program, shows the sensor data, as well as its value after encoding, and its value after decoding, to ensure that the encryption algorithm works correctly, and also shows the status of the lights, as well as the IP address of the internal network.



```
..
WiFi connected.
IP address:
192.168.43.43
Temperature = 33.50 Humidity = 55.20
Distance: 143.63

Original plain text for Temperature: 33.50
Ciphered text for Temperature: 9e3d11db20c1209dfb259f3ba2d11768
Deciphered text for Temperature: 33.50

Original plain text for Humidity: 55.20
Ciphered text for Humidity: 72b077da2ad2384d874e8ba18fe0badb
Deciphered text for Humidity: 55.20

Original plain text for Distance: 143.63
Ciphered text for Distance: 9e268e6e06397b151e73c276d05cfc9d
Deciphered text for Distance: 143.63     □□□?
New Client.
GET /5/on HTTP/1.1
Host: 192.168.43.43
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Figure 15. Result in serial arduino.

- Figure 16, shows the receipt of sensor data from a web page that can be accessed from any site in the world by the person who has access. The data that appears on this page may be limited to encrypted data, or other data may be added to it.



```
moh-esp32-1
https://moh-esp32-1-default-rtdb.firebaseio.com/

moh-esp32-1-default-rtdb
├─ Air Temp: 27
├─ "Distance: ": 15
├─ Gpio 4: "ON"
├─ Gpio 5: "oFF"
├─ Humidity: 38
├─ encryptDist: "4eff6e5152637b2bc35b776d69fa1872"
├─ encryptHumi: "0ab58ec44f60af3e15b5bd12c2849ad7"
├─ encryptTemp: "420c343f86c59b159e595547cea5924d"
└─ ip: "192.168.43.43"
```

Figure 16. Data sensors on Firebase web page.

- Figure 17, shows receipt of data on a website “Things peak .The validity of the encrypted data was verified by applying the AES algorithm on the Internet as a site that implements this algorithm after entering the data necessary for encryption.

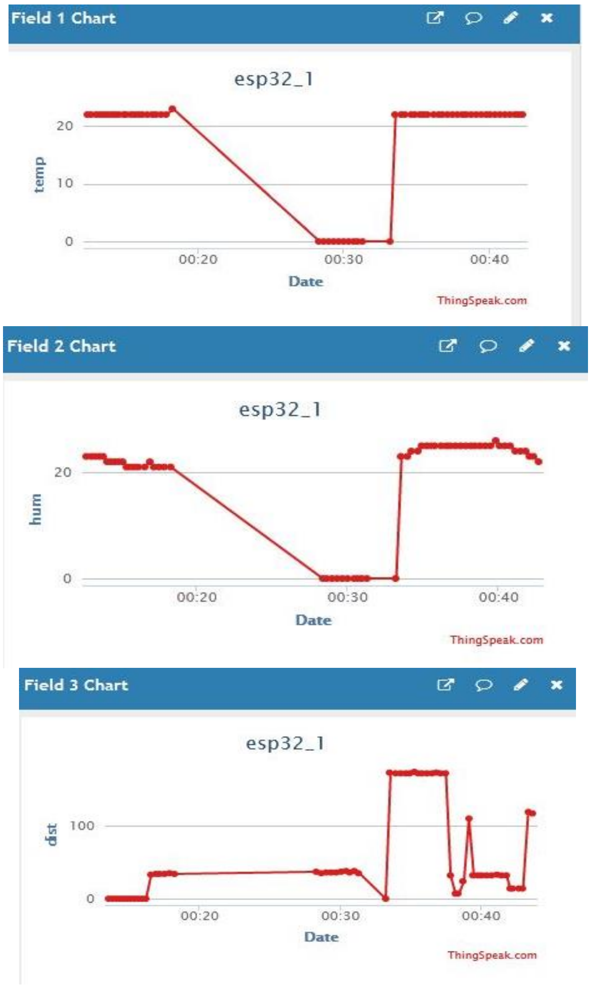
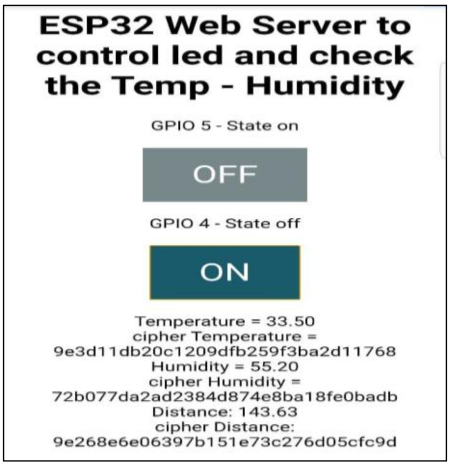


Figure 17. Data sensors on things peak web page.

- Figure 18, shows the receipt of the original value of the sensor data on the mobile device, and it also displays its value after encryption, and the ability to control the illumination of two lights. This interface is accessed via IP.



ESP32 Web Server to control led and check the Temp - Humidity

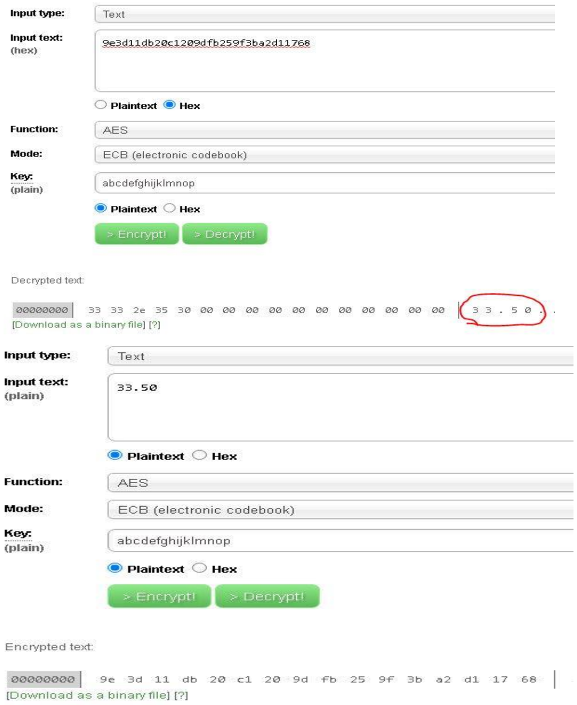
GPIO 5 - State on

GPIO 4 - State off

Temperature = 33.50
cipher Temperature =
9e3d11db20c1209dfb259f3ba2d11768
Humidity = 55.20
cipher Humidity =
72b077da2ad2384d874e8ba18fe0badb
Distance: 143.63
cipher Distance:
9e268e6e06397b151e73c276d05cfc9d

Figure 18. Data sensors on mobile.

- In Figure 19, we have verified the authenticity of the original and encrypted data received through a special application for the AES algorithm to demonstrate the design idea, which is the receipt of data in an encrypted form and then it is entered through an application with the key used in encryption to obtain the original values of the data, to keep it from being stolen. And to ensure the confidentiality of information.



Input type: Text

Input text: (hex) 9e3d11db20c1209dfb259f3ba2d11768

○ Plaintext ● Hex

Function: AES

Mode: ECB (electronic codebook)

Key: (plain) abcdefghijklmnop

● Plaintext ○ Hex

Decrypted text:

```
00000000 | 33 33 2e 35 30 00 00 00 00 00 00 00 00 00 00 00 | 3 3 . 5 0 .
```

[Download as a binary file] [?]

Input type: Text

Input text: (plain) 33.50

● Plaintext ○ Hex

Function: AES

Mode: ECB (electronic codebook)

Key: (plain) abcdefghijklmnop

● Plaintext ○ Hex

Encrypted text:

```
00000000 | 9e 3d 11 db 20 c1 20 9d fb 25 9f 3b a2 d1 17 68 |
```

[Download as a binary file] [?]

Figure 19. Crypt and decrypt of sensor data through an online application.

6. Conclusions

In this design, a mechanism was implemented that works to enhance and strengthen the security of the Internet of Things, by using the ESP32 platform to implement the encryption algorithm AES, and the design was also implemented on some sensors to represent the IoT part of the smart home or any other application. We chose the temperature, humidity, distance and control sensor with two lights.

This design can be applied to protect and secure the incoming data from the Internet of Things, due to what the ESP32 chip provides from dealing with IoT, as well as the strength provided by AES technology in protecting and securing the data received or sent.

References

[1] Abd Zaid M. and Hassan S., "Modification Advanced Encryption Standard for Design Lightweight Algorithms," *Journal of Kufa for Mathematics and Computer*, vol. 6, no. 1, pp. 21-27, 2019.

[2] Abdullah A., "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data Cryptography and Network Security," *Ryptography and Network Security*, vol. 16, pp. 1-11, 2017.

[3] Choi I. and Kim J., "Area-Optimized Multi-Standard AES-CCM Security Engine for IEEE 802.15. 4/802.15," *Journal of Semiconductor Technology and Science*, vol.16, no. 3, pp. 293-299, 2016.

[4] Chowdhury M., Ferdous M., Biswas K., Chowdhury N., Muthukkumarasamy V., "A Survey on Blockchain-Based Platforms for IoT Use-Cases," *The Knowledge Engineering Review*, vol. 35, 2020.

[5] "ESP32 Series Datasheet," Espressif Systems, Version 3.6, 2021.

[6] "ESP-IDF Programming Guide," https://docs.espressif.com/projects/esp-idf/en/latest/esp32/, Last Visited, 2020.

[7] Florin R. and Ionut R., "FPGA Based Architecture for Securing Iot with Blockchain," *in Proceedings of International Conference on Speech Technology and Human-Computer Dialogue*, Timisoara, pp. 1-8, 2019.

[8] Hamzah H., Ahmad N., and Ruslan S., "The 128-Bit AES Design by Using FPGA," *Journal of Physics: Conference Series*, vol. 1529, no. 2, pp. 022059, 2020.

[9] https://components101.com/sensors/dht11-temperature-sensor#, Last Visited, 2020.

[10] https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet, Last Visited, 2020.

[11] Hussein N. and Shujaa M., "DNA Computing Based Stream Cipher for Internet of Things Using MQTT Protocol," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp.1035, 2020.

[12] Inamdar A., "ESP32-S2-Security Features," *The ESP Journal*, https://medium.com/the-esp-Journal/esp32-s2-security-improvements-5e5453f98590, Last Visited, 2020.

[13] Khoa T., Nhu L., Son H., Trong N., Phuc C., Phuong N., Dung N., Nam N., Chau D., and Duc D., "Designing Efficient Smart Home Management with Iot Smart Lighting: A Case Study," *Wireless Communications and Mobile Computing*, vol. 2020, pp.1-18, 2020.

[14] Kodali R. and Soratkal S., "MQTT Based Home Automation System Using ESP8266," *in Proceedings of IEEE Region 10 Humanitarian Technology Conference*, Agra, pp. 1-5, 2016.

[15] Kouicem D., Bouabdallah A., and Lakhlef H., "Internet of Things Security: A Top-Down Survey," *Computer Networks*. vol. 141, pp. 199-221, 2018.

[16] Nandhini P. and Vanitha V., "A Study of Lightweight Cryptographic Algorithms for IoT," *International Journal of Innovations and Advancement in Computer Science*, vol. 6, no. 1, pp. 26-35, 2017.

[17] Parida D., Behera A., Naik J., Pattanaik S., and Nanda R., "Real-time Environment Monitoring System Using ESP8266 and Thing Speak on Internet of Things Platform" *in Proceedings of International Conference on Intelligent Computing and Control Systems*, Madurai, pp. 225-229, 2019.

[18] Postulka J., "Programming of ESP32 Microcontrollers," 2020.

[19] Raghavan R., "https://acodez.in/data-encryption-Algorithms/," and https://blog.storagecraft.com/5-common-Encryption-Algorithms/, Last Visited, 2021.

[20] Selmane N., Guilley S., and Danger J., "Practical Setup Time Violation Attacks on AES," *in Proceedings of 7th European Dependable Computing Conference*, Kaunas, pp. 91-96, 2008.

[21] Shanmuganathan H. and Mahendran A., "Encryption based on Cellular Automata for Wireless Devices in Iot Environment," *The International Arab Journal of Information Technology*, vol. 18, no. 3, pp. 347-355, 2021.

[22] Vacha M., "IoT Device Security on the ESP32 platform," Master's Thesis, Czech Technical University in Prague Computing and Information Center, 2020.

[23] Zhang J., Gao W., Li J., Tian X., and Dang H., "High-Speed and High-Security Hybrid AES-ECC Cryptosystem Based on FPGA," *in Proceedings of IEEE International Conference on Signal, Information and Data Processing*, Chongqing, pp. 1-6, 2019.

Mohammad Al-Mashhadani graduate student (Master) in Middle Technical University, Electrical Engineering Technical College. Iraq-Baghdad. majoring in Computer Technology Engineering-Holds a bachelor’s degree in the same specialty in (2004).

Mohamed Shujaa Graduate of the Polytechnic university of Bucharest, Faculty of Electrical and Computer Engineering (Computer Division)-PhD in Neural Network, Polytechnic university of Bucharest (2003), BSc. Electrical & Computer Engineering /Polytechnic University BUC. (1996), Absolvent of O levels degree from British academic /Nairobi (1978), Absolvent of A levels degree from British academic /Nairobi (1988).