Journal of Basrah Researches (Sciences) **49**(2), 30 (2023)
DOI: https://doi.org/10.56714/bjrs.49.2.4

Smart Building Security using ESP32 based AES One Bio-key and Owner's Biometrics Encryption Technology

Batool M. Radhi, Mohammed A. Hussain[*]

Department of Computer Science, Education College for Pure Sciences, University of Basrah, Basrah Iraq

ARTICLE INFO

Received	05 June 2023
Accepted	02 August 2023
Published	30 December 2023

Keywords:

ESP32, AES, Wi-Fi, *IoT* Security, Key generate, Biometrics.

Citation: B.M. Radhi, M.A. Hussain, J. Basrah Res. (Sci.) **49**(2), 30 (2023). DOI:https://doi.org/10.56714/bjrs.49.2.4

ABSTRACT

Internet of Things (IoT) connects smart devices to the internet, but it poses significant security challenges. ‘ Security measures like access control, authentication, and encryption must be implemented during IoT system development to safeguard sensitive information and ensure secure transmission. Regular security updates, ongoing assessment, and improved measures are necessary to address evolving threats. Data security is crucial for overall IoT system security, given its unique features like perception, transmission, and processing. In this paper, it is suggested using encryption for data sent over Wi-Fi to improve Internet of Things (IoT) security. Advanced Encryption Standard (AES) commonly used encryption standard, and it is the encryption technique employed. AES algorithm uses one bio-key (variable key every minute) to maximize security and the capacity to fend off attacks. It is added the owner's biometric to the bio-key for authentication and security in order to authenticate the system. Our technology stands out with robust encryption and authentication against numerous assaults because the owner's biometrics are encrypted and it is added to the bio-key made up of the minute-by-minute changes to the year, month, day, hour, and minute. The data is encoded in ESP32 and sent to the local server and then stored in a Python language database using the HTTP protocol. The database is hosted on XAMPP, which is a web server package that allows easy database management.

1. Introduction

The Internet of Things (*IoT*) is one of the most important modern technologies that are attracted the most interesting areas of life. Whether industrial, academic, or other. The Internet of Things (*IoT*) uses in the daily operation of many industries, applications and including but are not limited to,smart cities, smart grids, smart homes, physical security, e-health, asset management, and logistics. For example, the concept of smart city is emerging in multiple continents, where enhanced street lighting controls, infrastructure monitoring, public safety and surveillance, physical security, gunshot detection, meter

[*]**Corresponding author email:** mohammed.abdulridha@uobasrah.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https:// jou.jobrs.edu.iq

reading, and transportation analysis and optimization systems are being deployed on a city-wide scale[1].In recent years, the number of connected devices in the Internet of Things is increased tremendously. Whereas in 2015, nearly 15 billion devices are connected, and in 2019, they reached nearly 26 billion, and the number may reach about 75 billion devices by 2025 IoT refers to the interconnected network of physical devices, vehicles, home appliances, and other objects that are embedded with sensors, software, and network connectivity, enabling them to collect and exchange data over the internet[2].The devices in IoT ecosystem can range from simple sensors that monitor the temperature in a room to complex machinery used in manufacturing processes. The goal of IoT is to enable smarter and more efficient processes by automating tasks and providing real-time insights into data that is previously difficult to collect and analyze. This technology has the potential to revolutionize many aspects of our daily lives, including healthcare, transportation, energy management, Smart home, etc. [3].Smart building has become more and more popular in recent years. It aims at helping people manage the home appliance freely and build an autonomous environment in home or work area. A smart building is one that uses technology to enable efficient and economical use of resources, while creating a safe and comfortable environment for its occupants. Smart buildings may use a wide range of existing technologies and be designed or modified in a way that allows for the integration of future technological developments. Internet of Things (IoT) sensors, building management systems, and augmented reality are some of the mechanisms and robots that can be used in a smart building to control and improve its performance[4].When it is considered embedded IoT solutions, security is balanced between the three factors - Cost, Benefit and Risk. ‘Cost’ is the price for the design of security measures into industrial applications on the urgency of time to market. The ‘benefit’ of integrated security is the immediate access to critical features such as secure connections, authentication encrypted data storage, access-controlled ports, secure software updates, etc. Finally, the term 'risk' relates to the following: Hackers do not require physical access to equipment such as USB outlets or network ports when using remote and dispersed wireless networks, making distant industrial applications even more vulnerable to communication assaults[5].There are many threats and attacks affecting the Internet of Things and smart buildings, and this puts them at risk of security and privacy, Security and privacy are the two main goals for the Internet of Things. The main purpose of network security and Protect information to achieve confidentiality and safety. To provide security to IoT systems, such as confidentiality, integrity and authentication, the solution is in using an appropriate cryptographic algorithm. There are many cryptographic systems are used to provide security services, which are classified into symmetric and asymmetric types. But, most of the traditional cryptosystems cannot be used for secure IoT environments, because IoT works with constrained devices that limit battery, power, as well as memory. Hence, the design of efficient and lightweight cryptographic techniques became a challenge to guarantee secure data transmission in IoT networks. The lightweight cryptography is fitted the low energy, computation and memory capabilities of cyber-physical systems. Also, it is provided an optimized security/cost/performance trade-off [6]. When connecting a vast network of devices, this interconnectedness has also introduced new security and privacy challenges. With the increasing number of devices connected to the Internet, the risk of cyber-attacks and unauthorized access to sensitive data is a major concern. To address these security and privacy issues, various methods are being used to enhance the protection of data transmitted over the IoT network [7].Authentication is a crucial aspect in smart buildings to ensure both user authentication and data confidentiality. User authentication can be achieved through various methods such as passwords, biometric technology, or fingerprints[8]. Therefore, it has been implemented an authentication system to address this requirements[9].One such method is the use of encryption algorithms like AES[10] to encrypt data before it is transmitted. However, even this method has its limitations, as AES can still be vulnerable to brute force attacks, where attackers try to guess the encryption key through trial and error. Therefore, it is important to continue researching and developing security solutions to resist these attacks and eliminate loopholes in the AES algorithm using randomness and dynamic key[11] to ensure the safe and secure use of the Internet of Things[12][13] [14].

Our contributions

- The utilization of the bio-key in encryption is a significant contribution that greatly enhances the system's strength and its capacity to withstand numerous attackers. This is demonstrated in detail within the security analysis section.
- The use of the owner's biometrics contributes to the authentication of the system and satisfies the bio-key to safety and security of the system.
- Using ESP32 microcontroller with high technology in terms of processing speed and reasonable price compared to some high-cost devices.
- The use of Python language in research and is increases the execution speed due to the ready-made tools provided by the Python
- It is used a local server to store the data in a database after decrypting it, and this contributes to filtering the data, removing the junk data, and storing the data it is needed, whereas reduce the backup information.

2. Related Work

Al-Mashhadani and Shujaa (2022) [15], A group of sensors is suggested to be linked to the ESP32. The ESP32 chip receives and encrypts the data, which is then transmitted to a secure internet page that requires a username and password for access. However, there are some drawbacks to this approach. The encryption key is fixed, which makes it vulnerable to piracy. Additionally, the special internet page that requires a username and password can also be hacked(He used the owner's biometric for authentication). Lastly, this system is programmed using the "C" programming language. Sukiatmodjo and Setianto (2019) [16] found the sensors are connected by 2 Arduinos. Each Arduino is programmed in the C language using DES and AES algorithms, and the encrypted data is sent to a server that decrypts it using PHP. The results show that increasing the number of sensors leads to higher power consumption, and that the AES algorithm is faster in terms of encryption time. However, some disadvantages of the study include the use of Arduino instead of newer options such as ESP8266 or ESP32, and the use of C and PHP, which are more complex compared to Python. The owner did not use biometrics for authentication in the web server. Sikandar et al. [17], In the prototype, the researchers attempted to implement MQTT on ESP8266, a Wi-Fi-based development board. Sensors and actuators are connected to ESP8266, and a Mosquitto-based MQTT broker is established for remote monitoring and control. An MQTT client application is built on ESP8266, and the sensors and actuators connected to it are remotely monitored and controlled through a common home gateway. This implementation allows for the use of existing infrastructure to enhance home appliances and make them smart. The result is an intelligent, comfortable, and energy-efficient home automation system. It also assists the old and differently abled persons to control the appliances in their home in a better and easier way. Disadvantages in the study, the data transfer is not encrypted betit is en the broker, the client and the publisher, and this makes the system insecure.The researcher used ESP8266 although ESP32 is better and faster[18][15]. Yan et al. (2015) [19], design a smart home system with the implementation of related software and hardware. A set of sensors called the Smart Unit transmits data to a home proxy, which can be connected to a mobile phone or tablet. It is a console and also connected to the remote server via the internet. The combination of remote server and home proxy is a new scheme for remote control in which XMPP is used. Use home proxy to solve the synchronization problem and the system supports multiuser. It is also one phone that can register different Home proxies, thus one phone can control more than one smart home system or smart office system. So the remote server can provide services for different homes and offices . The disadvantage of the proposed system is the insecurity of the exchanged data because it is not encrypted. Chowdhury et al. (2019) [20], The Launchpad TI-CC3200 is a model that comes with a built-in microcontroller and an internal Wi-Fi shield, providing WPA or IT IS P security that allows for the control and management of electrical devices within the home. Additionally, the home security system can be operated without requiring the user to enable a data connection on their phone. The Launchpad is connected to a Wi-Fi network within the home or office, and the microcontroller can make decisions or send videos or photos to the owner or guest for further action. However, a disadvantage of the proposed system is that the board has WEP or WPA security, both of which have been compromised, rendering the TI-CC3200 Launchpad insecure [21]. T. A. Khoa Tran et al. in 2020[22], The system includes lighting and security monitoring features, and is designed to be used on a it is bsite or application. Each light bulb contains an ESP8266-12F microcontroller circuit, which receives data from sensors and sends that data to a it is b server. The it is b server displays information about the light bulbs and allows users to adjust the colors and turn the lights on or off. In automatic mode, the system uses data from a motion sensor to turn the lights on or off, and can also transmit a warning or siren alarm if the security mode is activated. The system functions as a security manager for the house, and users can integrate additional sensors into the system without having to install or configure the device. The system employs hardware components, a server, the it is b, and a mobile app. To ensure secure interactions betit is en multiple users and devices, the system uses a server based on SHA-256. Hoit is ver, there are some disadvantages to the proposed system. It uses an ESP8266 chip for each light bulb, which can be very expensive. Additionally, it employs only the simplest types of authentication SHA-256, and the transmitted data is not encrypted, exposing it to potential security risks. M. S. Fadhil et al. (2021) [6], In the present paper, the lightweight AES algorithm is used in providing security to IoT networks. lightweight AES (LAES) is a modification on AES algorithm layers, such as S-Box, Keys, and Shifting values, to be based on different chaotic systems. The IoT hardware components used in this work are Raspberry Pi device and sensors. The aim of this work is to protect IoT sensors data, such as temperature, humidity, and flame fire sensors, by the encryption process using the LAES algorithm before sending these data through the network. The LAES algorithm layers are different from the original AES in the following stages: The Initial Permutation (IP) is used instead of the Shift RowS and the dynamic Shift Rows is used instead of Mix Columns. Also, the SBox used in SubBytes operation is generated depending on the chaotic logistic map system. The disadvantage of research is the use of expensive Raspberry Pi and less storage compared to the ESP32 chip, in addition, the change occurred in the AES algorithm, this change is possible to know and hack, it is better if the change in the key is dynamic,and the owner's biometrics are not used in the local server for authentication.Savaştürk et al. (2021)[23], ESP32-CAM used in the study. The ESP32-CAM has the ability to capture both video and images, as well as connect to wireless networks for communication purposes. The ESP32-CAM encrypts video or images using the AES algorithm, and after encryption, the data is sent to the it is b server and from there to the desired device, such as a mobile phone, tablet, or computer. However, the problem is that the AES algorithm can be broken by Brute force attack[24]. To solve this issue, the algorithm can be made with a bio-key instead of a static one. Additionally, the researcher can use a local server instead of a it is b server to store data in a database and filter out unwanted data, which can further enhance the system's security, it did not use the owner's biometrics in the it is b server for authentication There are several disadvantages that it is found in the research that it is tried to overcome in our proposal, either by encrypting the data, using a dynamic bio-key, or by including authentication via biometrics of the owner as described in Algorithms 1,2,3

3. Problem Statement

The Internet of Things (IoT), which connects various systems, appliances, and devices, has revolutionized both our personal and professional life. However, this connectivity also generates new worries about security and privacy. There is a significant risk of cyberattacks and unwanted access to private data due to the rise in Internet-connected gadgets. Several techniques are being utilized to improve the protection of data carried over IoT network in order to address these problems. Applying encryption methods like AES to safeguard data before transmission is one such method[24]. AES has some limitations despite this strategy because attackers can still use brute force attacks to try and guess the encryption key. In addition to MITM attacks, there are eavesdropping and replay attacks as shown in part Security analysis. To maintain the use of the Internet of Things, it is must always research and develop confidentiality and authentication technologies [6].

4. Proposed Method

This paper presents a proposal to enhance the security of the Internet of Things (IoT) by implementing Wi-Fi data encryption. The chosen encryption method is AES, a widely adopted standard. However, considering the potential vulnerabilities of AES, it is introduce an additional security measure by utilizing a variable bio-key. This one bio-key is generated based on the current year, month, day, and minute, and it is synchronized with the local server for each transmission. By incorporating the owner's biometrics, the system achieves authentication and reinforces the security provided by the bio-key. The owner's biometrics are encrypted and combined with the bio-key, further strengthening the proposal. The encrypted data is stored in a database accessible through the Flask library in Python, employing the HTTP protocol. The database is hosted on XAMPP, a it is b server package that facilitates convenient database management. The architecture of the system is illustrated in Figure.1, highlighting its components and their interactions.

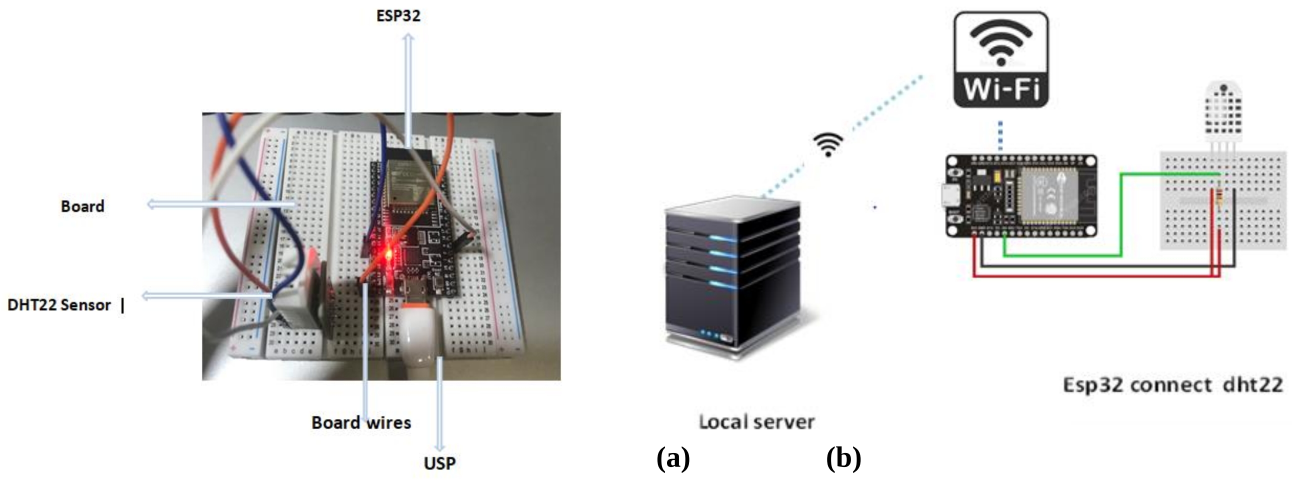


Fig. 1. Proposed design for secure and authenticated data transmission

Our design shows that the sensor communicates with the ESP32 chip via Wi-Fi, and the chip encrypts and authenticates (the owner's biometrics) the data in it, and then transmits it to the local server for decryption and storage.

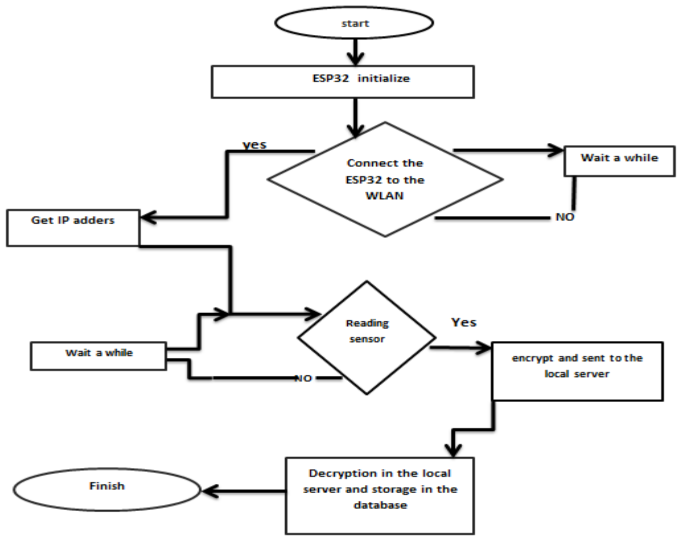

start
ESP32 initialize
Connect the ESP32 to the WLAN
yes
NO
Wait a while
Get IP adders
Reading sensor
Yes
NO
Wait a while
encrypt and sent to the local server
Decryption in the local server and storage in the database
Finish

Fig. 2. Structure showing system design. A structure showing the connection of the ESP32 chip to a WLAN, encrypting the data in it, and then transmitting it to the local server for decryption and storage.

The original encrypted data is received by implementing AES algorithm using a bio-key so that the key is the same for both ends. In this way, the data that is read, transmitted and stored in the database can be protected to prevent theft and ensure the confidentiality of the information. The algorithm is characterized by its strong encryption, and to increase its security, its key is variable, and it is found its high speed in encryption and decryption, and this is shown in Figure 8. If it is wanted to compare it with DES algorithm, one of the symmetric encryption algorithms, its key is smaller than AES, which is a little slower. Thus, it will be safer and faster to operate it as a reference [11]. This is one of the security and confidentiality features needed by the Internet of Things for smart buildings.

```
Algorithm 1. The owner's biometrics
Let  owner's biometrics  = $ow - bio$
Input    Fingerprint image(F img)
Output   Hash features fingerprint($ow - bio$)
    1. Procedure
    2. We used the opencv library in Python to read  F img
    3. Extract features  F img  using a canny algorithm
    4. Convert  Strong features   into a string
    5. Encrypt the result into a hash(sha1)
    6. End  $ow - bio$
```

```
Algorithm 2:  Encryption in $ESP32$
Let k = Bio-key 128-bit (16), $P$ = plaintext 128-bit, $C$ = ciphertext 128-bit, $dht22$ = sensor reading, T = temperature, $H$ = humidity.
Owner biometric = $ow - bio$
Input    dht22
Output   C
Phase 1.  Key generation and addition  Owner biometric
    1. ESP32  connected  to WLAN
    2. $dht22$ send to $ESP32$
    3. Generate $k$ dynamic by time exchange every     minutes
    4. generate the owner's biometrics and add them to the key
    5. $k = str(year[0] +_{+month[1]} +_{+day[2]} +_{+hour[3]} +_{+mint[4]} + ow - bio)$

Phase 2.  Encrypting data and sending it to the local server
 1. While true
         Try
                 $dht22 = T$ and $H$
                 $p = str(T) +,+ str(H)$
                 $Start\ time$ = Calculation of encryption time
                 $Cipher = AES\ (k, MODE\_CBC, iv)$
                 $C = iv + Cipher$
                 $Endtime = End\ of\ encryption\ time$
                 Request post( $HTTP:// IP: 5000$)
         Except Exception:
                 pass
```

```
**Algorithm 3 : Decryption in local server**
Let $DB$ = Database
 Environment flask app
Input $C$ (cipher text)
Output P(plaintext store in $DB$)
**Phase 1. Key generation and addition Owner biometric**
1. Procedure
2. $gettime$ (date time now)
Generate $k$ dynamic by time exchange every mint
3.generate the owner's biometrics and add them to the key
4. $K = str( y$
```

It is created three algorithms to illustrate the proposed method each algorithm works, as shown in Algorithm 1, Algorithm 2, and Algorithm 3 as following:

- **In Algorithm 1**. In the initial algorithm, it is retrieved the owner's biometric data by capturing their fingerprint and extracting its distinctive features. The Canny algorithm is employed for this feature extraction process. To secure the extracted robust features, it is utilized the SHA-1 hash function for encryption. The owner's biometric information is essential for authenticating the system and enabling its functionality in conjunction with the bio-key. See Algorithm 1.
- **In Algorithm 2**. The second algorithm focuses on encrypting the data and its transmission over Wi-Fi. Initially, in phase 1 we establish a connection between the ESP32 chip and the local network of the smart building. DHT22 readings are then transferred to the chip, as depicted in Figure 1. To ensure data security, the data is encrypted within ESP32 using AES algorithm in phase2 . Prior to encryption, a bio-key is generated, incorporating the year, month, day, hour, and minute. As the key changes every minute, this approach enhances the system's security. Additionally, the owner's biometric key is included to achieve system authentication. Following the encryption process, the data is transferred to the local server. See Algorithm 2.
- **In Algorithm 3**. The third algorithm is called the decryption algorithm after receiving the encrypted data. It generates a key from phase1 and the owner's biometric is added to it. In phase 2, here the process of decoding the code takes place and testing the validity of the data. If it is valid, it is stored in the data base. If it is not understood, there is a difference in time between the chip and the server,

so we work to generate another key one minute less. And it is test the validity of the data, if it is valid, it is stored in the database, and not crossed by an attacker. See Algorithm 3.

5. Implementation

Before connecting DHT22(temperature and humidity sensor) to ESP32, switch the ESP32 firmware to MicroPython instead of C programming. There are some differences in the libraries betit is en Python and MicroPython on ESP32. Once the sensor is connected, readings from it are sent to ESP32 chip, where they are encrypted using AES algorithm with a bio-ke . This key depends on the year, month, day, hour and minute and changes every minute with the inclusion of the owner's biometric to increase the strength of the system to counter many attacks. The encrypted data is then transmitted to the local server through WLAN using an IP address and HTTP protocol. The server then decrypts the data and stores it in a MySQL database. As shown in the Figure 2. For more details, see Pseudo Code.

Pseudo code

1. Generate a 32 bytes key

```
makeDynamicKey():
 key = getTime()+"95b6019e5548811b"
while len(key) < 32 :
    key += "#"
return bytes(key, "ASCII")
```

2. Sensor data is 16 bytes long, it is used random IV in AES encryption, calculate the encryption time, and send the encrypted data to the server

```
while len(resu) < 16:
        resu += " "
     iv = uos.urandom(BLOCK_SIZE)
    cipher = aes(makeDynamicKey(), MODE_CBC, iv)
    cipherText = iv + cipher.encrypt(resu)
    print(cipherText)
   endTime = time.localtime(time.time() + UTC_OFFSET)
   print(endTime - startTime)
   response = urequests.post("http://192.168.0.107:5000", data=cipherText)
  time.sleep(2)
```

3. In this code, the validity of the data sent from the chip is checked by generating a real-time key and adding the owner’s biometric and using it for decryption. If the data is valid, it is stored in the database

```
getTime():
  now = datetime.now()
  result =  now.strftime("%Y-%m-%d-%H-%M")
  return result
def makeDynamicKey():
  key = getTime()+"95b6019e5548811b"
  while len(key) <32 :
    key+="#"
  return bytes(key,"ASCII")
   databaseManager = mysql.connection.cursor()
    if(dataValid(request.data)):
      databaseManager.execute("INSERT    INTO    test    (text,test)
VALUES(%s,%s)",(finalResult,1))
      mysql.connection.commit()
      databaseManager.close()
      return "Done"
   else:
      print("Error")

    result = str(decodedText[:2]).isnumeric()
```

4. In this code, another new key is generated a minute less than the old key to verify the synchronization of the chip and the server, add the owner’s biometric, and decrypt the data. If it is valid, it is stored in the database, otherwise it is considered hacker in this case.

```
   key = makeDynamicKey()                                #String 2023-01-31-19-55
    lastNumber = int(key[14:16])
    lastNumber -= 1 ;
    StrLastNumber = str(lastNumber)
    checkTowNumber =makeTowNumbers(StrLastNumber)
    finalKey = key[:14]+bytes(checkTowNumber,"ASCII")   #String 2023-01-31-19-54
    cipher = AES.new(finalKey ,AES.MODE_CBC, iv)
    decryptedText = cipher.decrypt(cipherText)[BLOCK_SIZE:]
    decodedText = decryptedText.decode("latin-1")
    result = str(decodedText[:2]).isnumeric()
```

5.1 Project Structure Components

A. Sensor(dht22)
B. Server(local server)
C. Esp32 chip

Table 1. DHT22 sensor Specifications

Model	DHT22
Poit is r supply	3.3-6V DC
Output signal	digital signal via single-bus
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Long-term Stability	+-0.5%RH/year
Humidity hysteresis	+-0.3%RH
Interchangeability	fully interchangeable

A. Sensor: DHT22

One of the main components of this project is the DHT22 sensor humidity and Temperature (show in Figure 3). Table 1 shows DHT22 Specifications [25].

B. Server (local server) A server is a computer system that provides resources, data, services, or programs to other computers or devices on a network. A local server is a server that is located on the same network as the clients that access it, as opposed to a remote server which is located in a different location and is accessed over the Internet. Local servers are commonly used for tasks such as file sharing, printing, and hosting local applications or websites. They can also be used to centralize the management of data and services within an organization, making it easier to access and share information among employees [6].

C. ESP32 Chip (Overview ESP32)

The ESP32 is a 2.4GHz Wi-Fi and Bluetooth chip manufactured using 40nm ultra-low-poit is r TSMC technology. This is to accomplish the greatest performance in addition to Radio Frequency (RF) performance, and is distinguished by dependability, longevity, and adaptability in a wide range of poit is r situations and applications. 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 4 [26][27].

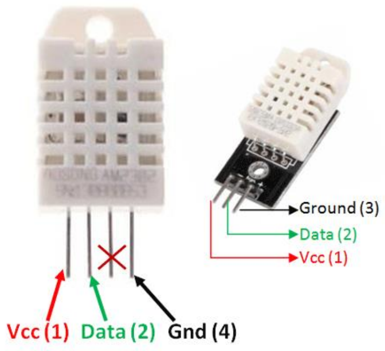


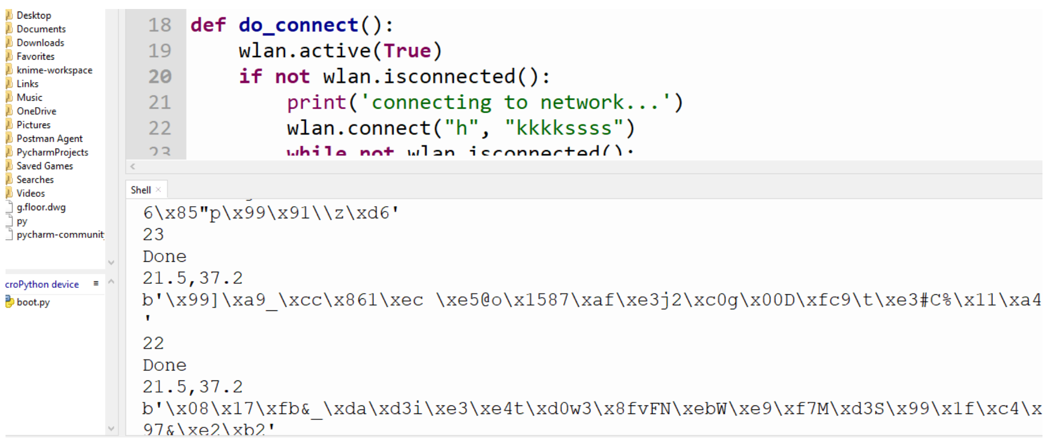
Fig.3. DHT22 sensor (sensor humidity and Temperature)

Fig.4. Esp32 WROOM-32S (ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip).

6. Result and Discussion

Design shown in Figure 1, Start by opening the database in XAMPP and open the local server , sensor readings are normal temperature and humidity readings (e.g. 25.33,70.25).The data is received from the sensors via the ESP32, in which the 128-bit AES algorithm and bio-key is used(y-m-d-h-m)with owner's biometrics(e.g.2023-03-20-06-25 +jgffetfdyhgfdkjhg) To encrypt the data and send it to the local server which will decrypt the data and store it in a XAMPP database that can be referenced when needed.

Fig.5 Represents a shell at Thonny shows the encoded sensor data along with the time it takes to encode this data. To ensure that the encryption algorithm is working correctly, quickly and safely.

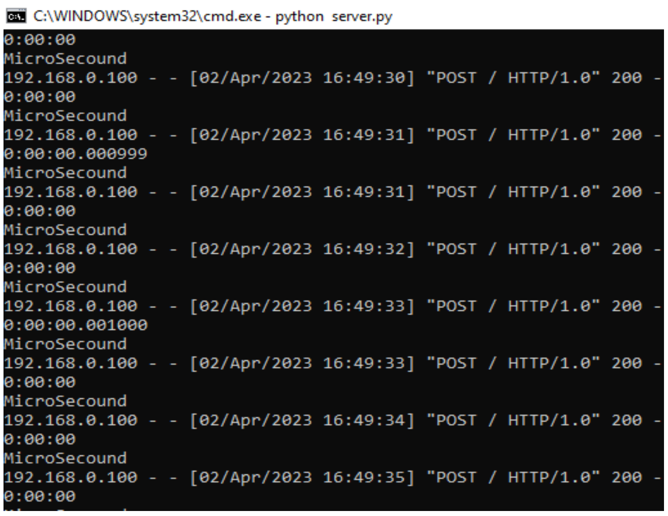


```
def do_connect():
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        wlan.connect("h", "kkkkssss")
        while not wlan.isconnected():
```

```
6\x85"p\x99\x91\\z\xd6'
23
Done
21.5,37.2
b'\x99]\xa9_\xcc\x861\xec \xe5@o\x1587\xaf\xe3j2\xc0g\x00D\xfc9\t\xe3#C%\x11\xa4
'
22
Done
21.5,37.2
b'\x08\x17\xfb&_\xda\xd3i\xe3\xe4t\xd0w3\x8fvFN\xebW\xe9\xf7M\xd3S\x99\x1f\xc4\x
97&\xe2\xb2'
```

Fig.5. Result in shell (The data in the chip is encrypted with the owner's biometric and sent to the local server)

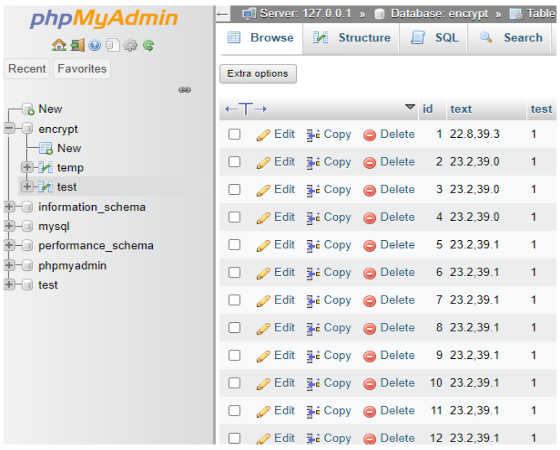
In Fig.6 CMD displays the local server program open to receive encrypted sensor data from ESP32 via the chip's IP. The data decoding process occurs on the server while calculating the decoding time to ensure the accuracy of the data before storing it in the database.



```
C:\WINDOWS\system32\cmd.exe - python  server.py
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:30] "POST / HTTP/1.0" 200 -
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:31] "POST / HTTP/1.0" 200 -
0:00:00.000999
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:31] "POST / HTTP/1.0" 200 -
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:32] "POST / HTTP/1.0" 200 -
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:33] "POST / HTTP/1.0" 200 -
0:00:00.001000
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:33] "POST / HTTP/1.0" 200 -
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:34] "POST / HTTP/1.0" 200 -
0:00:00
MicroSecound
192.168.0.100 - - [02/Apr/2023 16:49:35] "POST / HTTP/1.0" 200 -
0:00:00
```

Fig.6. Result Data sensors on local server (The local server receives the encrypted data, decrypts the data, and stores it in DB)

Fig.7. shows the stored sensor readings in the database, presented in plaintext format.



	id	text	test
Edit Copy Delete	1	22.8,39.3	1
Edit Copy Delete	2	23.2,39.0	1
Edit Copy Delete	3	23.2,39.0	1
Edit Copy Delete	4	23.2,39.0	1
Edit Copy Delete	5	23.2,39.1	1
Edit Copy Delete	6	23.2,39.1	1
Edit Copy Delete	7	23.2,39.1	1
Edit Copy Delete	8	23.2,39.1	1
Edit Copy Delete	9	23.2,39.1	1
Edit Copy Delete	10	23.2,39.1	1
Edit Copy Delete	11	23.2,39.1	1
Edit Copy Delete	12	23.2,39.1	1

Fig.7. Result Data sensors on database (stores the data in a database).

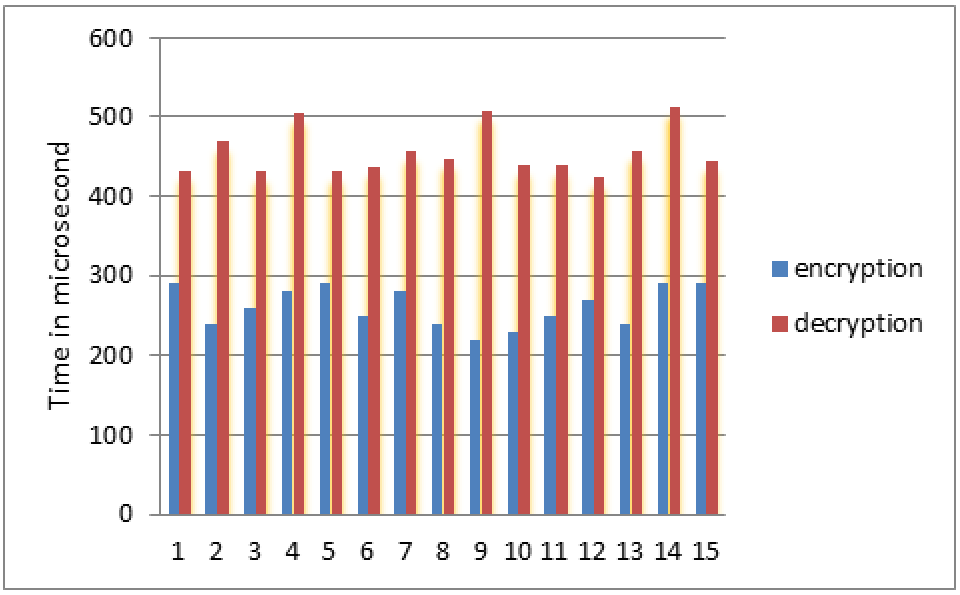


Fig.8. Encryption time comparison with decryption

(the time is calculated using an average time after executed several time).

The time delay in both encryption and decryption is shown in the Figure 8, where the time is calculated using an average time after executed several time. the decryption takes more time caused by the validation phase. Hoit is ver, the time is reasonable for both processes.

7. Security analysis

In this part it is discussed the strength of the scheme against some common attacks and the attempts of our system to overcome them.

1. **Theorem1.** The proposed system is designed to resist attacks by incorporating a Once bio-key and owner biometrics.

proof: In this system, AES encryption algorithm is utilized, and the encryption key is dynamically generated as a one bio-key. The key is updated based on the current variable time per minute, ensuring that it changes frequently. To enhance security further, the biometric features of the owner, specifically the characteristics of the owner's fingerprint, are extracted and added to the variable key. When encrypting data in the system, this modified key, which incorporates the owner's biometrics, is used. The encryption process takes place on ESP32 device, as described in Algorithm 1 and 2. Similarly, on the local server, the same AES algorithm and process are applied. For decryption, the process follows the steps outlined in Algorithm 1 and 3. By utilizing data encryption and authentication mechanisms, the system ensures the confidentiality and integrity of the transmitted data. This combined approach of incorporating a one-time bio-key and owner biometrics into the encryption process helps the system resist many attacks. The use of a dynamic key based on variable time and the addition of owner biometrics make it significantly more challenging for attackers to gain unauthorized access or manipulate the encrypted data.

2. **Theorem2**. Our proposed method can resist A brute force attack

Proof: Brute force attack[28] An attacker attempts to gain access to an encrypted system or data by systematically trying all possible combinations of passwords or encryption keys until the correct one is found. One of the key strategies is the use of a frequently changing bio-key. the proposed method By changing the bio-key every minute, it is ensured that even if the attacker attempts to guess the key, it is needed to start the guessing process anew with each change. This significantly increases the complexity and time required for a successful brute force attack. As shown in phase 1 (4-6) of the algorithm (2), algorithm (3).

3. **Theorem3**. Our proposed approach can resist A MITM (Man-in-the-Middle) attack

Proof: The attacker can eavesdrop on the connection, manipulate the data being transmitted, or even impersonate one or both of the parties involved. In our plan, it is implemented measures to resist MITM attacks[29] effectively. One of the key strategies is encrypting the data sent over Wi-Fi. By encrypting the data, it is ensure that even if an attacker intercepts it, they won't be able to understand or make use of the information without the decryption key. Encryption adds an extra layer of security to the communication process and mitigates the risk of unauthorized access. As shown in phase 2 of the algorithm (1). In addition, it is incorporated biometrics into our security procedures. By recording the owner's biometrics and using them together with a key, it is established a strong authentication mechanism. Biometric data, such as fingerprints or facial recognition, is unique to each individual, making it difficult for an attacker to impersonate authorized parties involved in the communication. As shown in phase 1 (5-6) of the algorithm (2) and phase (4-5) algorithm (3).

4. **Theorem4.** Our proposed approach can resist A replay attack

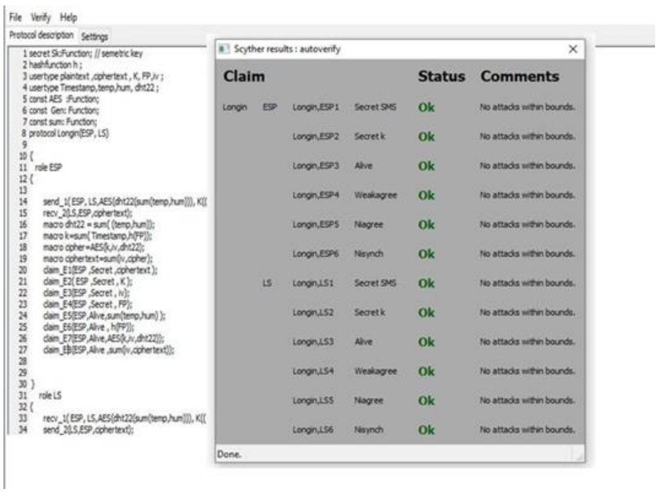
Proof: In our scheme, it is implemented measures to effectively confront replay attacks[30]. One of the key aspects is the encryption of the data with a bio-key that changes every minute. This encryption process ensures that even if an attacker manages to capture and replay the data packets, they will be ineffective as the data has changed due to the encryption with the bio-key. By utilizing a changing bio-key, it is add an extra layer of security to the transmitted data. The dynamic nature of the key means that any captured packets become outdated and unusable after a minute, rendering the replay attack ineffective. This feature distinguishes our scheme and provides robust protection against replay attacks.

5. **Theorem5.** Our proposed approach can resist An eavesdropping attack

Proof: also known as interception or snooping[31], The attacker's objective is to gain unauthorized access to sensitive or confidential information transmitted over the network. In our scheme, it is successfully countered this attack by implementing robust security measures. Specifically, it is employed the AES algorithm to encrypt the data transmitted betit is en the sensor and the local server, ensuring its confidentiality and integrity.

8. Scyther Tool

It can swiftly look into a protocol's properties and is a tool for the formal study of security protocols. Used to assess the security and weaknesses of schemes. The tool's operation can be broken down into two steps. Scyther promises to complete the initial phase while allowing an infinite number of sessions to confirm the reliability of the protocol[32]. The alternative is to provide the evidence (via the backend), and Scyther supports graphical interface analysis by providing a second-stage attack based on behavior categorization. The SPDL programming language is used to create Scyther protocols. SPDL supports a number of crucial cryptographic processes, including message sending and receiving between components and the roles that each component plays[33]. (See Figure **9).**

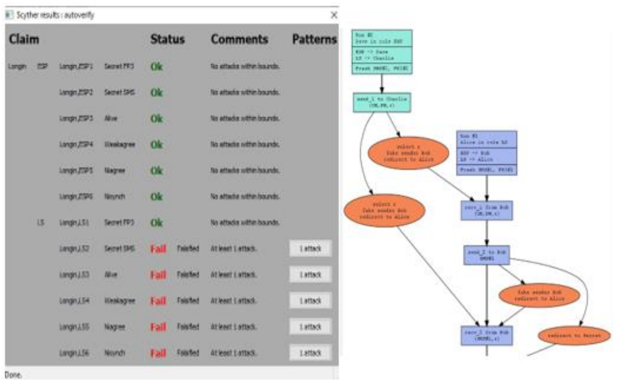


```
secret Sk:Function; // semetric key
hashfunction h ;
usertype plaintext ,ciphertext , K, FP,iv ;
usertype Timestamp,temp,hum, dht22 ;
const AES  :Function;
const  Gen: Function;
const sum: Function;
protocol Longin(ESP, LS)

{
  role ESP
{

    send_1( ESP, LS,AES(dht22(sum(temp,hum))), K{(
    recv_2(LS,ESP,ciphertext);
    macro dht22 = sum( (temp,hum));
    macro k=sum( Timestamp,h(FP));
    macro cipher=AES(k,iv,dht22);
    macro ciphertext=sum(iv,cipher);
    claim_E1(ESP ,Secret ,ciphertext );
    claim_E2( ESP ,Secret , K );
    claim_E3(ESP ,Secret , iv);
    claim_E4(ESP ,Secret , FP);
    claim_E5(ESP,Alive,sum(temp,hum) );
    claim_E6(ESP,Alive , h(FP));
    claim_E7(ESP,Alive,AES(k,iv,dht22));
    claim_E8(ESP,Alive ,sum(iv,ciphertext));



}
   role LS
{
    recv_1( ESP, LS,AES(dht22(sum(temp,hum))), K{(
    send_2(LS,ESP,ciphertext);
```

Scyther results : autoverify

Claim				Status	Comments
Longin	ESP	Longin,ESP1	Secret SMS	Ok	No attacks within bounds.
		Longin,ESP2	Secret k	Ok	No attacks within bounds.
		Longin,ESP3	Alive	Ok	No attacks within bounds.
		Longin,ESP4	Weakagree	Ok	No attacks within bounds.
		Longin,ESP5	Niagree	Ok	No attacks within bounds.
		Longin,ESP6	Nisynch	Ok	No attacks within bounds.
	LS	Longin,LS1	Secret SMS	Ok	No attacks within bounds.
		Longin,LS2	Secret k	Ok	No attacks within bounds.
		Longin,LS3	Alive	Ok	No attacks within bounds.
		Longin,LS4	Weakagree	Ok	No attacks within bounds.
		Longin,LS5	Niagree	Ok	No attacks within bounds.
		Longin,LS6	Nisynch	Ok	No attacks within bounds.

Done.



Scyther results : autoverify

Claim				Status		Comments	Patterns
Longin	ESP	Longin,ESP1	Secret FP3	Ok		No attacks within bounds.	
		Longin,ESP2	Secret SMS	Ok		No attacks within bounds.	
		Longin,ESP3	Alive	Ok		No attacks within bounds.	
		Longin,ESP4	Weakagree	Ok		No attacks within bounds.	
		Longin,ESP5	Niagree	Ok		No attacks within bounds.	
		Longin,ESP6	Nisynch	Ok		No attacks within bounds.	
	LS	Longin,LS1	Secret FP3	Ok		No attacks within bounds.	
		Longin,LS2	Secret SMS	Fail	Falsified	At least 1 attack.	1 attack
		Longin,LS3	Alive	Fail	Falsified	At least 1 attack.	1 attack
		Longin,LS4	Weakagree	Fail	Falsified	At least 1 attack.	1 attack
		Longin,LS5	Niagree	Fail	Falsified	At least 1 attack.	1 attack
		Longin,LS6	Nisynch	Fail	Falsified	At least 1 attack.	1 attack

Done.

(a) cannot be attacked
(b) can be attacked

Fig.9. Illustrates the results of the system scheme verification (Login and Authentication phase that cannot be attacked and can be attacked)

9. Conclusion

This paper introduces the concept of the Internet of Things, including its implementation in smart homes and smart buildings to enhance homeowners' quality of life. The study involved converting the ESP32 chip from the C programming language to MicroPython, and this increases the execution speed due to the ready-made tools provided by the Python language. Readings are gathered by the DHT22 sensor and passed to the ESP32 chip for processing and encryption. To be stored in a database, the encrypted data is transferred through HTTP to a local server. The AES method is used in conjunction with a bio-key and the owner's biometrics (authentication) to achieve the highest levels of security and confidentiality. The key dynamically changed every minute during encryption in the ESP32 and transmission to the server. This method maintains a high level of security and authentication while enabling quick encryption and decryption. it can be stored only the information when it is needed by filtering the data as needed.

10. References

[1] D. Minoli, K. Sohraby, B. Occhiogrosso, in IEEE Internet of Things Journal, **4** (1), 269(2017). Doi: https://doi.org/10.1109/JIOT.2017.2647881

[2] R. Florin, R. Ionut, 2019 10th Int. Conf. Speech Technol. Human-Computer Dialogue, SpeD 2019, 1(2019). Doi: https://doi.org/10.1109/SPED.2019.8906595

[3] A. Yousefi, S. M. Jameii, IEEE Int. Conf. IoT its Appl. ICIOT, 3(2017). Doi: https://doi.org/10.1109/ICIOTA.2017.8073627

[4] P. M. B. Mansingh, G. Sekar, Adv. Eng. Res. 49, 197(2022).

[5] S. Krishnan, M. S. Anjana, S. N. Rao, 2017 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2017, 1(2018). Doi: https://doi.org/10.1109/ICCIC.2017.8524450

[6] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, Iraqi J. Sci., **62**(8), 2759(2021), Doi: https://doi.org/10.24996/ijs.2021.62.8.29

[7] A. Mosenia, no. Icoei, 454(2018).

[8] Z. A. Abduljabbar et al., Proc. - 2nd IEEE Int. Conf. Big Data Secur. Cloud, IEEE BigDataSecurity 2016, 2nd IEEE Int. Conf. High Perform. Smart Comput. IEEE HPSC 2016

IEEE Int. Conf. Intell. Data Secur. IEEE IDS 2016, 146(2016). Doi: https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.25
[9] V. Kumar, N. Malik, J. Singla, N. Z. Jhanjhi, F. Amsaad, A. Razaque, Cryptography, **6**(3), 2022. Doi: https://doi.org/10.3390/cryptography6030037
[10] A. Muhammad Abdullah, no. June, 2017. [Online].
[11] V. O. Nyangaresi, Z. A. Abduljabbar, Z. A. Abduljabbar, 2021 IEEE 2nd Int. Conf. Signal, Control Commun. SCC 2021, 188 (2021). Doi: https://doi.org/10.1109/SCC53769.2021.9768338
[12] V. O. Nyangaresi, M. A. A. Sibahee, Z. A. Abduljabbar, J. Ma, M. S. Khalefa, MELECON 2022 - IEEE Mediterr. Electrotech. Conf. Proc., 726(2022). Doi: https://doi.org/10.1109/MELECON53508.2022.9842900
[13] M. A. A. Sibahee et al., ICSPCC 2020 - IEEE Int. Conf. Signal Process. Commun. Comput. Proc., January, 2020. Doi: https://doi.org/10.1109/ICSPCC50002.2020.9259519
[14] S. Marksteiner, V. J. E. Jimenez, H. Valiant, H. Zeiner, Jt. 13th CTTE 10th C. Conf. Internet Things - Bus. Model. Users, Networks, 1(2017). Doi: https://doi.org/10.1109/CTTE.2017.8260940
[15] M. Al-Mashhadani, M. Shujaa, Int. Arab J. Inf. Technol., **19**(2), 214 (2022). Doi: https://doi.org/10.34028/iajit/19/2/8
[16] A. Sukiatmodjo Y. D. Setianto, Sci. J. Informatics, **6**(1), 45(2019). Doi: https://doi.org/10.15294/sji.v6i1.17838
[17] R. K. Kodali, S. Soratkal, 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 1(2016). Doi: https://doi.org/10.1109/R10-HTC.2016.7906845
[18] R. Khwanrit, S. Kittipiyakul, J. Kudtonagngam , H. Fujita, International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES), Khon Kaen, Thailand, 1(2018). Doi: https://doi.org/10.1109/ICESIT-ICICTES.2018.8442066
[19] W. Yan, Q. Wang, Z. Gao, Chinese Control Conf. CCC, 9072(2015). Doi: https://doi.org/10.1109/ChiCC.2015.7261075
[20] S. S. Chowdhury, S. Sarkar, S. Syamal, S. Sengupta, P. Nag, 2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019, 1158 (2019). Doi: https://doi.org/10.1109/UEMCON47517.2019.8992994
[21] C. P. Kohli's, T. Hayajneh, Electron., 7 (11), 2018. Doi: https://doi.org/10.3390/electronics7110284
[22] T. A. Khoa et al., Wirel. Commun. Mob. Comput., 2020. Doi: https://doi.org/10.1155/2020/8896637
[23] P. Savaştürk, Ö. Aydın, G. Dalkılıç, SSRN Electron. J., **17**(4), 447(2022). Doi: https://doi.org/10.2139/ssrn.4171323
[24] L. I. B. Mahendra, Y. K. Santoso, G. F. Shidik, Proc. - 2017 Int. Semin. Appl. Technol. Inf. Commun. Empoit is r. Technol. a Better Hum. Life, iSemantic 2017, 66 (2017). Doi: https://doi.org/10.1109/ISEMANTIC.2017.8251845
[25] Aosong Electronics, New York Aosong Electron., 22, 1(2015).
[26] M. Babiuch, P. Foltynek, P. Smutny, Proc. 2019 20th Int. Carpathian Control Conf. ICCC, 1 (2019). Doi: https://doi.org/10.1109/CarpathianCC.2019.8765944
[27] E. Systems, "ESP32 Series," 2023.
[28] J. Park, J. Kim, B. B. Gupta, N. Park, Comput. Mater. Contin., **68**(1), 887 (2021). Doi: https://doi.org/10.32604/cmc.2021.015172
[29] H. I. Nasser M. A. Hussain, ijeee.edu.iq, 8 (2023). Doi: https://doi.org/10.37917/ijeee.19.2.2.

[30] G. 2021. 3059648. pdfja. Suthokumar, V. Sethu, C. Wijenayake, E. Ambikairajah, Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, 691(2018). Doi: https://doi.org/10.21437/Interspeech.2018-1846

[31] T. M. Hoang, T. Q. Duong, H. D. Tuan, S. Lambotharan, L. Hanzo, IEEE Access, 9, 31595 (2021). Doi: https://doi.org/10.1109/ACCESS.2021.3059648

[32] H. A. Elbaz, M. H. Abd-elaziz, T. M. Nazmy, International Journal of Distributed and Cloud Computing, 2 (2),1(2014).

[33] N. Hamed, A. Yassin, Iraqi J. Electr. Electron. Eng., **18**(1),71 (2022). Doi: https://doi.org/10.37917/ijeee.18.1.9.

Journal of Basrah Researches (Sciences) **49**(2), 30 (2023)
DOI: https://doi.org/10.56714/bjrs.49.2.4

Smart Building Security using ESP32 based AES One Bio-key and Owner's Biometrics Encryption Technology

بتول محمد راضي، محمد عبد الرضا حسين *

قسم علوم الحاسوب، كلية التربية للعلوم الصرفة، جامعة البصرة، البصرة، العراق.

معلومات البحث | الملخص

الاستلام	05 أيار 2023
القبول	02 اب2023
النشر	30 كانون الأول 2023

الكلمات المفتاحية

ESP32 قطعة ، AES، Wi-Fi ، أمان إنترنت الأشياء ، توليد المفاتيح ، القياسات الحيوية.

إنترنت الأشياء ، يربط الأجهزة الذكية بالإنترنت ، لكنه يفرض تحديات أمنية كبيرة. لحماية المعلومات الحساسة وضمان النقل الآمن ، يجب تنفيذ تدابير أمنية مثل التحكم في الوصول والمصادقة والتشفير أثناء تطوير نظام إنترنت الأشياء. تعد التحديثات الأمنية المنتظمة والتقييم المستمر والتدابير المحسنة ضرورية لمواجهة التهديدات المتطورة. يعد أمان البيانات أمرًا ضروريًا لأمن نظام إنترنت الأشياء بشكل عام ، نظرًا لميزاته الفريدة مثل الإدراك والنقل والمعالجة. في هذه الورقة ، نقترح تحسين أمان إنترنت الأشياء (IoT) من خلال تنفيذ التشفير للبيانات المنقولة عبر شبكة Wi-Fi. طريقة التشفير المستخدمة هي معيار التشفير المتقدم (AES) ، وهو معيار تشفير مستخدم على نطاق واسع. بمجرد استخدام المفتاح الحيوي (المفتاح المتغير في الدقيقة) في خوارزمية AES لزيادة الأمان والقدرة على مقاومة الهجمات قدر الإمكان. لمصادقة النظام ، قمنا بتضمين القياسات الحيوية للمالك في المفتاح الحيوي للمصادقة والأمان. يتم تشفير القياسات الحيوية للمالك وإضافتها إلى المفتاح الحيوي المكون من السنة والشهر واليوم والساعة والدقيقة ويتغير كل دقيقة وهذا ما يميز نظامنا بتشفير قوي ومصادقة ضد العديد من الهجمات. يتم تشفير البيانات في ESP32 وإرسالها إلى الخادم المحلي ثم تخزينها في قاعدة بيانات لغة Python باستخدام بروتوكول HTTP. قاعدة البيانات مستضافة على XAMPP.

Citation: B.M. Radhi, M.A. Hussain, J. Basrah Res. (Sci.) **49**(2), 30 (2023). DOI:https://doi.org/10.56714/bjrs.49.2.4

*Corresponding author email: mohammed.abdulridha@uobasrah.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https:// jou.jobrs.edu.iq