



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.524

9940 572 462

6381 907 438

ijrset@gmail.com

www.ijrset.com

Ultra-Light-Weight Encryption for Securing D2D Communication of ESP32 IOT Devices in Wireless Mesh Networks

Dr. V.Senthilkumar¹, Anusha Gorre²

Associate Professor, Department of Electronics and Communication Engineering, Er.Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India ¹

PG Scholar, Department of Applied Electronics, Er.Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India²

ABSTRACT: This research focuses on enhancing the security of Device to Device (D2D) interaction in wireless mesh networks, specifically targeting ESP32 IoT devices. We propose the implementation of ultra-light-weight encryption algorithms to secure the communication channels, ensuring data integrity and confidentiality. The MATLAB platform is utilized for simulation and validation purposes. The ultra-light-weight encryption solution is tailored to the constraints of ESP32 devices, considering their limited computational capabilities and memory resources. Our approach prioritizes efficiency while maintaining a security, making it more suitable for real-world usage IoT deployments in wireless mesh networks. Through MATLAB simulations, we assess the effectiveness of the proposed encryption scheme represented by communication overhead, latency, overall system robustness. The results demonstrate the feasibility and effectiveness of the ultra-light-weight encryption for ESP32 IoT devices in securing D2D communication within wireless mesh networks. Create a simulation environment in MATLAB to model D2D communication in a wireless mesh network. Integrate the ESP32 devices with the implemented ultra-light-weight encryption. This framework provides a structured approach to implementing ultra-light-weight encryption for ESP32 IoT devices in D2D communication within wireless mesh networks using MATLAB.

KEYWORDS: MATLAB, ESP 32, Device-to-Device (D2D)

I. INTRODUCTION

Device-to-device (D2D) communication is a system where electronic devices interact directly without relying on an intermediary network infrastructure. It is prevalent in wireless communication, data sharing, and connectivity. D2D allows devices like ESP32 microcontrollers and IoT devices to establish direct connections, facilitating faster data transfer and reducing network infrastructure load. It is also relevant in peer-to-peer data sharing, as devices equipped with D2D capabilities can exchange information directly, bypassing centralized servers or cloud-based services. To enhance privacy, reduce data latency, and improve efficiency in file sharing, content streaming, and collaborative tasks. D2D communication implementation involves technologies like Bluetooth and Wi-Fi Direct. The widespread adoption of D2D communication has implications for developing decentralized and peer-to-peer applications, contributing to the evolution of modern communication systems.

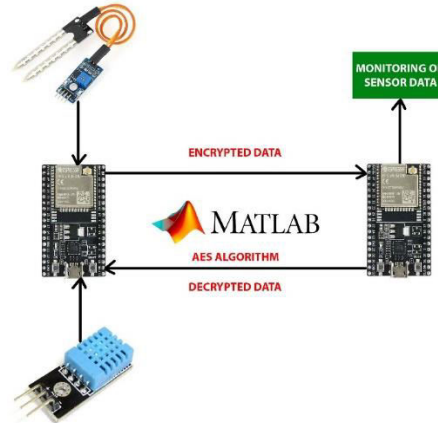


Figure 1 Encrypted Data Transfer between ESP32 Controllers

The importance of secure and efficient data transmission in digital communication has grown, particularly in Device-to-Device (D2D) communication. This technology allows devices to communicate directly without relying on centralized networks, but the need to protect sensitive information during D2D exchanges is growing. D2D data encryption is a solution to this issue, implementing cryptographic techniques to encode information between devices, ensuring only authorized parties can access and interpret it. This proactive approach safeguards against unauthorized access and strengthens the integrity and confidentiality of the exchanged information. D2D encryption is a safeguard for sensitive data and a key enabler for the evolution of decentralized communication systems.

The rise of digital communication and interconnected devices has made securing sensitive information crucial. Device-to-device (D2D) communication, a crucial element of the Internet of Things (IoT) involves ensuring robust encryption to secure data exchanged between devices. The Advanced Encryption Standard (AES) is a widely recognized and reliable encryption algorithm for device-to-device (D2D) encryption, working on fixed-size data blocks. AES ensures confidentiality and integrity by using a secret key known only to the communicating devices, which is used in both encryption and decryption operations. AES's strength lies in its high-security level while maintaining efficiency and computational speed, optimizing for resource-constrained devices in Device-to-Device (D2D) communication scenarios. This introduction focuses on the critical role of AES in D2D data encryption, emphasizing its reliability, versatility, and ability to tackle the changing challenges of safeguarding confidential data in the dynamic environment of interconnected devices.

1.1 Objective

To evaluate the performance of the encryption scheme using MATLAB simulations, focusing on communication overhead, latency, and system robustness.

To demonstrate the feasibility and effectiveness of the encryption solution for securing D2D communication within wireless mesh networks.

To create a simulation environment in MATLAB to model D2D communication in a wireless mesh network.

To integrate ESP32 devices with the ultra-lightweight encryption implemented in the simulation framework.

To provide a structured approach for implementing ultra-light-weight encryption for ESP32 IoT devices in D2D communication within wireless mesh networks using MATLAB.

II. LITERATURE SURVEY

Industrial applications, such as transportation, healthcare, and building energy management systems, rely on the fusion of wireless Ad-hoc/mesh networking and Key-Policy Attribute-Based Encryption (KPABE) parameters. The suggested unified system designed to provide lightweight security with KPABE for wireless mesh networking ensures adaptability and expandability. However, ensuring the seamless integration of KPABE into real-world applications necessitates meticulous alignment with hardware platforms and communication systems. The outstanding performance

in mesh networking and KPABE functionality is demonstrated, with future efforts focusing on enhancing the non-pairing KPABE scheme [1].

A security algorithm employing chaotic encryption of text messages through two tiers of one-dimensional chaotic maps with distinct initial conditions. Ensures 100% correct decryption of the original text without errors, and ensure the critical spacing surpasses the minimum key spacing to fend off brute-force assaults. Implemented using an Arduino microcontroller board (ATmega2560), the system uses two levels of encryption, generating pseudo-random numbers and increasing critical space. The system reduces microcontroller operating time and has a perfect real-time operation speed [2].

Designing an application-specific Instruction Set Processor (ASIP) for lightweight encryption, employing techniques such as the Data Encryption Standard (DES) and Advanced Encryption Standard (AES), ensure security. Comparing machine instruction composition and computational speed, AES outperforms DES in terms of fewer instructions and performance, with an average computational time of 0.0544653 seconds for separate I/O data scenarios and 0.0520902 seconds for data overwrite scenarios [3].

The COVID-19 pandemic has led to a surge in online activity, necessitating the development of secure encryption algorithms. The generation of pseudo-random numbers and the execution of multimedia encryption utilize chaotic map-based pseudo-random number generation techniques. The method has applications in various fields, including multimedia missile systems. Classic hardware such as FPGA, alongside contemporary controllers like Arduino and Raspberry Pi, have been utilized for this purpose. Different types of chaotic maps, such as logistic types, Tinkerbell maps, and Ikeda maps, have been employed for generating pseudo-random numbers. The Arithmetic Logic Unit (ALU) in microcontrollers is capable of processing the same number of bits as the random-number generators. While not ideal for real-time tasks, pseudo-random number generators based on chaotic maps are useful for generating OTPs and encrypting images [4].

The automotive sector is seeing a significant rise in vehicle-to-vehicle, vehicle-to-infrastructure, and vehicle-to-cloud communications, demanding increased data speeds for efficient network connectivity. Encryption plays a vital role in safeguarding data transmitted through these networks. The Advanced Encryption Standard (AES) is extensively utilized in automotive microcontrollers. Field-Programmable Gate Arrays (FPGAs) are employed for pre-silicon validation and software creation, facilitating the deployment of real-time encryption algorithms such as AES. A high-throughput FPGA implementation of the AES algorithm for automotive microcontrollers has been suggested, showcasing its efficacy in bolstering security and operational efficiency [5].

The foundational stratum of the Internet of Things (IoT) plays a pivotal role in upholding security, particularly concerning data gathering, scrutiny, and commands. A methodical safeguarding solution is advocated, employing an RC6 encryption/decryption algorithm implemented in MATLAB. This methodology tackles confidentiality and operational efficacy apprehensions within practical settings, guaranteeing that unauthorized entities are unable to decipher data. Assessments validate the steadfastness of the technique's encrypted data, affirming its robustness. Subsequent research endeavors ought to prioritize enhancing both stability and ubiquity in the security of the IoT's foundational layer [6].

Wireless networks and IoT devices play a vital role in overseeing smart environments, encompassing security surveillance and medical systems. Nonetheless, decentralized battery-operated sensors might heighten data latency and diminish energy efficiency. An innovative D2D multi-criteria learning algorithm is proposed to enhance data transmission without extra expenses and mitigate risks. This algorithm enhances packet delivery rate, minimizes packet interference, reduces data latency, conserves energy, and simplifies computational complexity for practical network setups. However, it remains vulnerable to link interruptions. Future studies ought to employ deep learning architectures and real-time datasets to address network irregularities [7].

The noteworthy advancement of IoT systems, amalgamating sensors, actuators, and communication apparatus, poses a substantial obstacle in guaranteeing security. The assurance of effective encryption stands paramount for IoT systems, wherein symmetric and asymmetric key encryption stand as pivotal types. This study introduces a peer-to-peer encryption algorithm, tailored explicitly for IoT applications, aimed at tackling the security hurdles linked with IoT systems [8].

The realm of the Internet of Things (IoT) confronts hurdles when crafting efficient and resilient cybersecurity protocols for devices limited in resources. This study suggests an indomitable digital signature fused with a pragmatic, efficient encryption (ELCD) system employing secure key distribution via elliptic curve Diffie-Hellman (ECDH). ELCD method verifies the sender's authenticity decisively while ensuring perfect forward secrecy. The security of ELCD has been formally substantiated utilizing random oracle and IoT adversary models, surpassing conventional cryptographic methods in terms of CPU execution time, storage expenses, and power usage. Future endeavors will concentrate on amplifying the efficiency of a compact digital certificate built upon ECDH within IoT frameworks [9].

The term "The Internet of Things (IoT)" refers to an expansive array of devices possessing constrained processing and communication abilities, all interconnected within 5G networks. Conventional backend systems encounter difficulties in upholding security and privacy due to the sheer volume of these devices. To tackle these challenges, the Cloud Approximate Nearest Neighbors (CANN) mechanism employs an upgraded variant of the Advanced Encryption Standard (I-AES) and a Privacy Database Structure (PDS). This approach effectively addresses security and privacy concerns. The proposed I-AES algorithm surpasses its predecessors in terms of encryption execution time and throughput [10].

III. PROPOSED METHODOLOGY

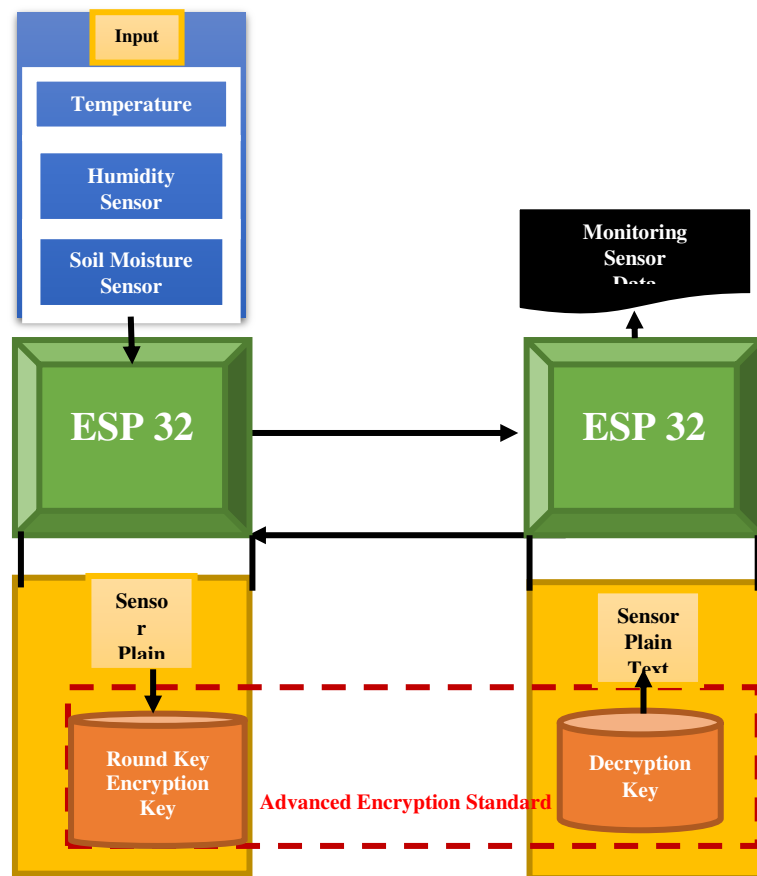


Figure 2 Proposed Block Diagram

The suggested approach prioritizes bolstering the security of device-to-device (D2D) communication within wireless mesh networks, particularly involving ESP32 microcontroller-based IoT devices. The system utilizes two ESP32 microcontrollers to establish a secure communication channel for exclusively sharing sensor data, such as humidity,

temperature, and soil moisture, between these two nodes. The communication is achieved through D2D communication, ensuring a direct and efficient data transfer mechanism. Two ESP32 microcontrollers (Node 1 and Node 2) were utilized to facilitate the direct communication of sensor data. This approach minimizes latency and reduces dependency on centralized networks, making it suitable for real-time data exchange applications. The nodes are configured to establish a dedicated link solely for sharing sensor information, ensuring a secure and efficient communication path. To safeguard the transmitted sensor data, the project utilizes the Advanced Encryption Standard (AES) algorithm for both encrypting and decrypting data.

The encode process involves generating an AES key, which is utilized to secure the sensor data before transmission. Encryption and decryption keys are crucial components of secure data transfer to ensure the data remains confidential and protected from unauthorized access during its transfer from Node 1 to Node 2. To implement this, MATLAB is employed to develop the key code, which is then integrated into the system using the C language. To ensure the creation of robust and secure encryption and decryption mechanisms, enhancing the overall reliability of the proposed system. The encrypted sensor data is transmitted from Node 1 to Node 2 and decrypted using the pre-shared AES key. This secure data transfer ensures the confidentiality and integrity of the transmitted information. The decrypted sensor data can then be monitored and utilized for various applications, such as environmental sensing, agriculture, or industrial automation. The chosen AES algorithm balances security and computational efficiency, making it well-suited for resource-constrained IoT devices like ESP32 microcontrollers.

Ultra-lightweight encryption ensures that the security measures do not impose significant overhead on the system's performance, making it feasible for real-world IoT applications. The proposed project introduces a secure communication framework for ESP32 microcontroller-based IoT devices using ultra-lightweight encryption. Implementing D2D communication and the robust AES algorithm provides a reliable and efficient solution for protecting sensor data during transfer. Integrating MATLAB for key code development further enhances the project's versatility. This secure communication system is poised to find applications in diverse fields, ensuring the integrity and confidentiality of data exchanged between IoT devices in wireless mesh networks.

3.1 Advanced Encryption Standard (Aes) Algorithm

The Advanced Encryption Standard (AES) stands as a prevalent symmetric encryption algorithm, serving to safeguard sensitive information. It was introduced by the National Institute of Standards and Technology (NIST) in 2001, aiming to supersede the Data Encryption Standard (DES). AES functions on data blocks and accommodates key lengths of 128, 192, or 256 bits. D2D communication is direct communication between nearby devices without a centralized network infrastructure. This type of communication is prevalent in various scenarios, such as peer-to-peer file sharing, collaborative applications, and Internet of Things (IoT) devices. AES can be employed in D2D communication for safe and secure data transfer by encrypting and decrypting collected data.

AES secures the data shared between devices, guaranteeing that only authorized entities possessing the accurate encryption key can decode and retrieve the data. This safeguards against unauthorized interception and eavesdropping. Additionally, AES is employed to generate cryptographic hash functions or Message Authentication Codes (MACs), ensuring the data's integrity remains intact during transmission and hasn't been tampered with. Proper management of encryption keys is imperative for D2D communication. AES offers flexibility with various key lengths, chosen based on the specific security needs of the application. It's essential to implement effective key exchange mechanisms to securely distribute and update encryption keys among devices, particularly in scenarios where resources may be limited. When employing AES in D2D communication, factors like key length should be carefully considered to strike a balance between security and computational efficiency. Consequently, integrating AES into D2D communication amplifies the security and confidentiality of exchanged data, fostering the establishment of reliable and resilient communication environments.

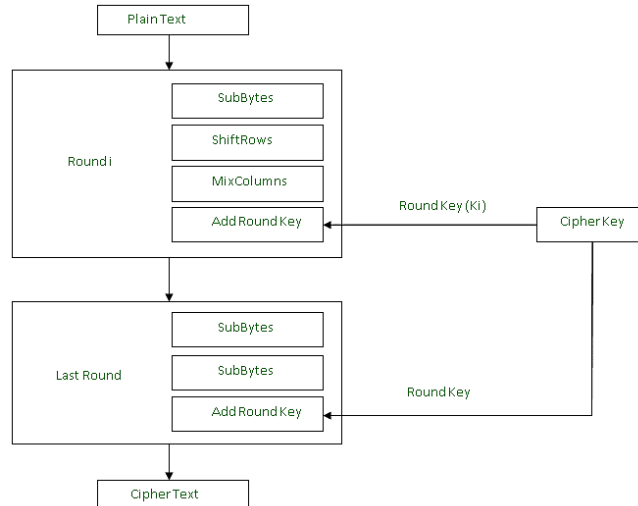


Figure 3 AES Encryption Process with XOR Operation and Round Key Integration

3.2 Encryption

Encryption is a method that guarantees only authorized individuals can access and comprehend confidential information. The Advanced Encryption Standard (AES) is a prevalent encryption technique, and MATLAB incorporates built-in functionalities for AES encryption. The encrypt function employs the AES algorithm to encrypt data and generates a secure key for both encryption and decryption processes. Prior to transmission, the sender (device A) utilizes the AES algorithm and a shared key to encrypt the data. AES divides the data into blocks and applies substitution, permutation, and mixing operations. To uphold confidentiality and integrity, establishing secure channels for data exchange is imperative in Direct-to-Distributed (D2D) communication. Upon reception, the receiver (device B) uses the identical key and AES algorithm to decrypt the data, restoring it to its original state.

3.3 Decryption

Deciphering involves transforming encoded information back into its initial, comprehensible state. Within D2D (Device-to-Device) interaction, deciphering plays a pivotal role in upholding the confidentiality and integrity of exchanged data. AES (Advanced Encryption Standard) stands as a prevalent symmetric encryption technique in D2D interaction and various other contexts. It operates on data blocks, employing a concealed key for both encryption and decryption processes. AES's hallmark lies in its symmetrical nature, where the identical key serves for both operations. In D2D interaction utilizing AES, deciphering entails reversing the encryption process by applying the AES algorithm to the encoded data. The very same covert key employed during encryption is utilized to revert the encoded data to its original, unencrypted state. Ensure that only devices with the correct secret key can decrypt and access sensitive information.

3.4 Round Key in AES Algorithm

Decoding involves converting encoded data back into its original, understandable form. In D2D (Device-to-Device) communication, decoding is crucial for maintaining the privacy and accuracy of shared information. AES (Advanced Encryption Standard) is a widely used symmetric encryption method in D2D communication and various other applications. It functions by operating on data blocks and employing a secret key for both encryption and decryption. AES's key feature is its symmetrical nature, where the same key is used for both encryption and decryption. In D2D communication employing AES, decoding involves reversing the encryption process by applying the AES algorithm to the encoded data. The same secret key used during encryption is employed to restore the encoded data to its original, unencrypted form.

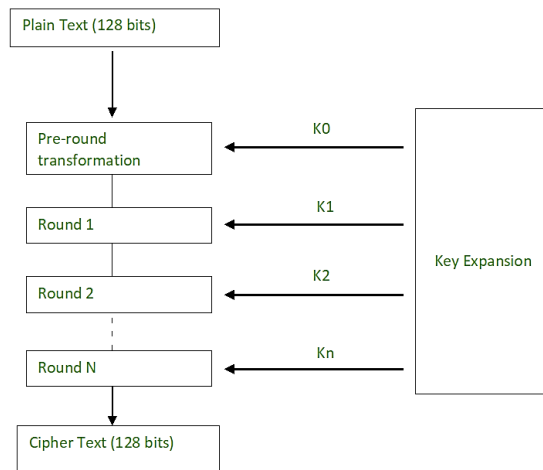


Figure 4 Round Key in AES Algorithm

3.5 Hardware Components

A. ESP 32 MICROCONTROLLER

The ESP32 represents a highly adaptable microcontroller crafted by Express if Systems. Renowned for its utility in Internet of Things (IoT) scenarios, this device boasts a dual-core processor, integrated Wi-Fi, Bluetooth capabilities, generous GPIO pins, and compatibility with diverse communication protocols. Its utilization of the Advanced Encryption Standard (AES) ensures the safeguarding of device-to-device (D2D) data exchanges. Notably, the ESP32 outperforms single-core counterparts owing to its dual-core Tensilica Xtensa LX6 processor, delivering enhanced performance and efficiency.

In terms of connectivity, the ESP32 supports both Wi-Fi (802.11 b/g/n) and Bluetooth, catering to a wide spectrum of IoT ventures reliant on wireless communication. Furthermore, it is equipped with an array of peripherals, encompassing GPIO pins, SPI, I2C, and UART, enabling seamless interfacing with myriad sensors, actuators, and peripherals. Engineered with a focus on power efficiency, it emerges as a fitting choice for applications emphasizing battery longevity and energy conservation.

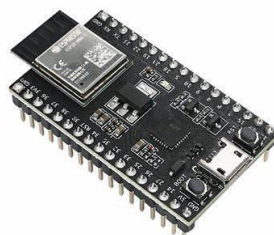


Figure 5 ESP 32 Microcontroller

Security constitutes a pivotal element in IoT applications, and the ESP32 tackles this challenge by incorporating features like hardware-accelerated encryption (AES, SHA-2), secure boot functionality, and a robust random number generator. The ESP32 seamlessly integrates with the Arduino IDE, ensuring accessibility to a diverse community of developers. It boasts support for both classic Bluetooth and Bluetooth Low Energy (BLE), broadening its applicability across various domains.

The ESP32 thrives on the active and supportive ecosystem of open-source contributors, fostering collaboration and the exchange of knowledge. Both ESP32 devices must undergo initialization with the AES algorithm, encompassing key

generation and distribution. Encrypted data is transmitted between these devices utilizing a communication protocol such as Wi-Fi or Bluetooth, contingent upon the device's capabilities and the specific requirements of the application. Upon receipt of the encrypted data, the recipient device employs the shared secret key to decrypt it, restoring it to its original state.

B DHT 11 SENSOR

The DHT11 sensor stands as a widely favored and cost-effective digital sensor for detecting both temperature and humidity across a myriad of electronic domains and applications. It boasts the capability to gauge temperatures within the span of 0°C to 50°C (32°F to 122°F) with an error margin of $\pm 2^\circ\text{C}$, while also accurately measuring relative humidity ranging from 20% to 80% with a deviation of $\pm 5\%$.

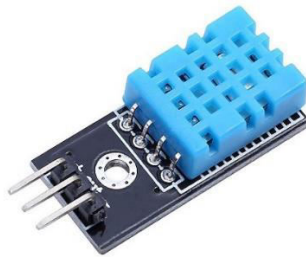


Figure 6 Diagram of DHT11 Sensor

The sensor offers digital output, making its connection with microcontrollers or other digital devices simpler. Inside the sensor module, the DHT11 sensor integrates a microcontroller and signal-processing circuitry. Its function includes converting analog sensor readings into digital signals. Notably, the DHT11 exhibits a somewhat sluggish response time, averaging around 2 seconds for temperature and 2-5 seconds for humidity.

C SOIL MOISTURE SENSOR

Soil moisture detectors play a crucial role in agriculture, horticulture, and environmental surveillance by furnishing up-to-the-minute information on soil moisture levels. They are instrumental in refining irrigation techniques, safeguarding water reservoirs, and boosting plant development. These detectors gauge soil moisture by evaluating the electrical resistance amid electrodes, which escalates with increasing soil moisture, thereby influencing the capacitance amid electrodes. The dielectric constant of soil undergoes alterations with varying moisture levels, thereby impacting the capacitance amid electrodes. Soil moisture levels can also be gauged by weighing a soil sample pre and post-drying, and by measuring soil moisture tension or suction. Detectors are typically inserted into the soil at diverse depths, contingent on the root zone of the plants under observation. Data loggers or interconnected systems can relay sensor data for real-time surveillance.



Figure 7 Diagram of Soil Moisture Sensor

3.6 Software Description

MAT LAB SOFTWARE DESCRIPTION

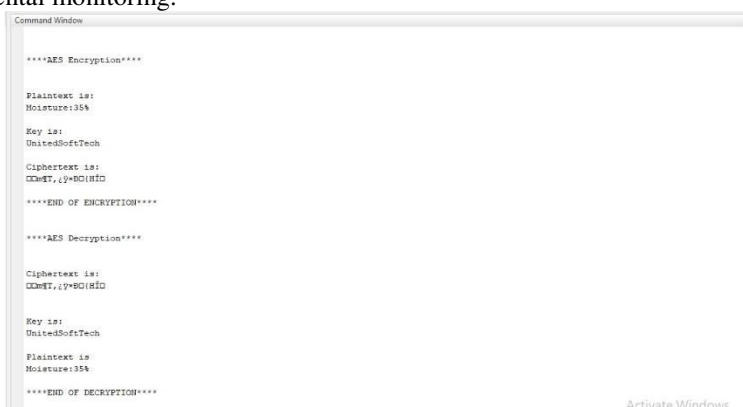
MATLAB is a powerful tool for technical computing, offering computation, visualization, and programming in a user-friendly environment. Originally focused on matrix computation, it has evolved into a versatile language with extensive graphics capabilities and interfaces with other languages like C and FORTRAN. Its toolboxes provide specialized solutions for various domains. The language supports both small-scale programming for quick solutions and large-scale programming for complex applications.

Key features include:

- 1.Array Pre-Allocation: This ensures efficient memory usage by pre-allocating arrays.
- 2.JIT Acceleration: Improves the speed of M-functions, particularly with loops.
- 3.Data Types: supports various data types for acceleration, with some limitations.
- 4.Multithreaded Computation: Enhances performance on multi-core and multiprocessor systems.
- 5.Desktop Tools and Development Environment: Offers a comprehensive set of tools for productive MATLAB usage.
- 6.Simulink provides a block diagram environment for multi-domain simulation and model-based design.
- 7.Modeling: Allows the modeling of algorithms and physical systems using block diagrams.
- 8.Block Libraries: Offers a wide range of predefined blocks for building models.
- 9.Simulation: Supports interactive simulation and batch simulations with MATLAB integration.
- 10.Performance: Provides tools like Performance Advisor for optimizing simulation speed.
- 11.Component-Based Modeling: Enables modular design and independent verification of components.
- 12.Modeling Guidelines: Offers guidelines for improving the consistency, clarity, and readability of models.
- 13.Block Creation: Allows the creation of custom blocks using MATLAB functions, enhancing modeling flexibility.

IV. RESULT AND DISCUSSION

The proposed system simulates soil moisture sensor data using MATLAB, encrypting it using the Advanced Encryption Standard (AES) algorithm. The figure 4.1 shows output of the simulation indicates a moisture level of 35%. The AES algorithm secures sensitive data using a secret encryption and decryption are facilitated by a cryptographic key. The data is then transmitted between controllers, ensuring confidentiality and integrity. Encryption converts data into a protected form that is only accessible by using the correct key for decryption. The system's security relies on properly managing encryption keys, with each controller having the key for decryption. To ensure authorized entities can access and interpret the transmitted soil moisture data. The system's security relies on properly managing encryption keys, ensuring only authorized entities can access and interpret the data. This approach is designed to safeguard the integrity and confidentiality of soil moisture information, which could be crucial in applications like agriculture or environmental monitoring.



```
Command Window

****AES Encryption****

Plaintext is:
Moisture:35%

Key is:
UnitedSoftTech

Ciphertext is:
CmE7,2*%01HID

****END OF ENCRYPTION****

****AES Decryption****

Ciphertext is:
CmE7,2*%01HID

Key is:
UnitedSoftTech

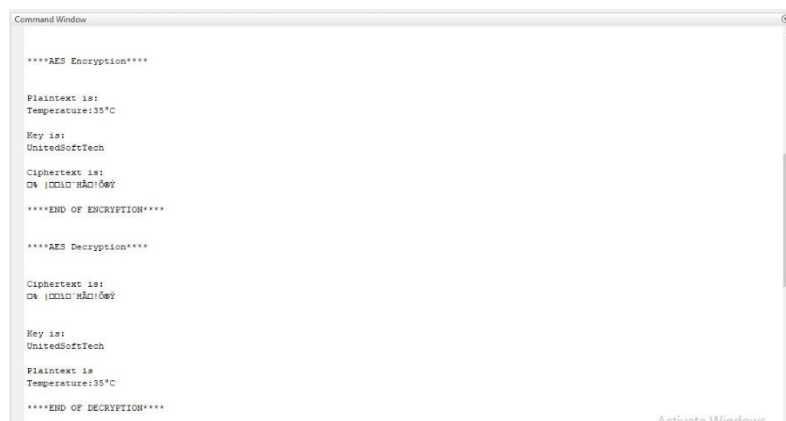
Plaintext is:
Moisture:35%

****END OF DECRYPTION****

Activate Windows
```

Figure 8 Simulation Output of Moisture Sensor

A temperature sensor is simulated using MATLAB; the figure 4.2 represent the simulation output, equivalent to a temperature of 35 degrees Celsius, is processed using MATLAB. The MATLAB simulation output is encrypted and decrypted using the AES algorithm, which is crucial for securing sensitive data like temperature data. Encryption converts the data into an unintelligible form, whereas decryption undoes this process to retrieve the initial data. This encryption is essential for sensitive information like temperature data, as it protects against unauthorized access or tampering during data transmission. The encryption and decryption processes require a key, a parameter the AES algorithm uses to perform these operations. The goal is to achieve safe and secure data transmission between controllers, ensuring only authorized controllers can access and interpret the information and ensuring confidentiality and integrity of the transmitted data.



```
Command Window

****AES Encryption****

Plaintext is:
Temperature:35°C

Key is:
UnitedSoftTech

Ciphertext is:
DN |Cm|n H&I|0e?

****END OF ENCRYPTION****

****AES Decryption****

Ciphertext is:
DN |Cm|n H&I|0e?

Key is:
UnitedSoftTech

Plaintext is
Temperature:35°C

****END OF DECRYPTION****
```

Figure 9 Simulation Output of Temperature Sensor

A humidity sensor simulation output is represented in Figure 4.3, a moisture level of 75%, likely corresponding to a specific environmental condition. MATLAB, a widely used programming and numerical computing platform, generates the simulation output. The Enhanced Encryption Protocol (EEP) is utilized on the synthesized output information for the purpose of encoding and decoding, safeguarding the data from unauthorized interception during transit. Encoding and decoding operations hinge upon a distinct passphrase, guaranteeing exclusive access to the initial data solely for authorized entities. The passphrase plays a pivotal role in both encoding and decoding processes. The ultimate goal is to achieve safe and secure data transmission between controllers by encrypting the information before transmission and decrypting it at the receiving end using the appropriate key, protecting the integrity and confidentiality of the data being communicated between different controllers.



```
Editor - D:\RBD\Krishnakumar\AES_MATLAB-master\AES_MATLAB-master\AES_MAT\aes_master.m
Command Window

****AES Encryption****

Plaintext is:
Humidity:75%

Key is:
UnitedSoftTech

Ciphertext is:
DN_Bu 700D.CDlK

****END OF ENCRYPTION****

****AES Decryption****

Ciphertext is:
DN_Bu 700D.CDlK

Key is:
UnitedSoftTech

Plaintext is
Humidity:75%

****END OF DECRYPTION****
```

Figure 10 Simulation Output of Humidity Sensor

V. CONCLUSION

In conclusion, a comprehensive and secure communication framework tailored for ESP32 IoT devices engaged in Device-to-Device (D2D) communication within wireless mesh networks. The primary focus is on enhancing the security of sensor data transfer, particularly in applications like agriculture, environmental monitoring, and industrial automation. The proposed framework employs ultra-light-weight encryption algorithms, explicitly emphasizing the Enhanced Encryption Protocol (EEP) for safeguarding the secrecy and reliability of crucial information. The suggested unique contribution lies in its integration of MATLAB for simulation, key code development, and validation, showcasing a systematic approach to implementing ultra-light-weight encryption for ESP32 devices. The proposed encryption scheme was tested using MATLAB simulations to evaluate its communication overhead, latency, and system robustness.

The results confirm the feasibility and effectiveness of the ultra-light-weight encryption, demonstrating its applicability to real-world IoT deployments. The security protocols of the system encompass the replication of soil moisture, temperature, and humidity sensors. Each category of data undergoes encryption and decryption procedures utilizing the AES algorithm. Encryption keys serve as crucial components in preserving the confidentiality and integrity of transmitted data, thereby permitting solely authorized parties to access and comprehend the information. The suggested technology addresses the inherent challenges of securing D2D communication in wireless mesh networks with resource-constrained IoT devices but also presents a versatile and practical solution for applications where data integrity and confidentiality are paramount. The proposed secure communication framework, backed by MATLAB simulations and ultra-light-weight encryption, holds promise for diverse fields requiring reliable and protected data exchange among IoT devices.

REFERENCES

1. Hijawi, U., Unal, D., Hamila, R., Gastli, A. and Ellabban, O., "Lightweight KPABE Architecture Enabled in Mesh Networked Resource-Constrained IoT Devices," *IEEE Access*, VOL. 9, pp.5640-5650, 2020.
2. Msekh, Z. A., & Hreshee, S. S., "Implementation of a chaos-based symmetric text encryption using Arduino microcontrollers," In *Journal of Physics: Conference Series*, vol. 1963, no. 1, pp. 012086, 2021.
3. Karna .N, S. Febriani, R. Nugraha and D. -S. Kim, "Performance Analysis on x86 Architecture Microprocessor for Lightweight Encryption," *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, pp. 262-265, 2020.
4. Naik, R.B., Singh, U, "A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption. Ann." *Data. Sci*, 2022.
5. Sikka, P., Asati, A.R. & Shekhar, C, "High-throughput field-programable gate array implementation of the advanced encryption standard algorithm for automotive security applications," *J Ambient Intell Human Comput*, vol. 12, pp. 7273–7279, 2021.
6. Ashraf. A. M., W. M. Elmedany and M. S. Sharif, "Secure IoT Data Transmission at Physical Layer using RC6 Encryption Technique," *9th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 307-315, 2022.
7. Haseeb, Khalid, Amjad Rehman, Tanzila Saba, Saeed Ali Bahaj, and Jaime Lloret. "Device-to-Device (D2D) Multi-Criteria Learning Algorithm Using Secured Sensors," *Sensors*, vol. 22, no. 6, pp. 2115, 2022.
8. Saikia, M., "An efficient D2D quaternion encryption system for IoT using IEEE 754 standards," *Internet of Things*, vol. 11, pp.100261, 2020.
9. Ahmed, Adel A., and Omar M. Barukab. "Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cyber security Mechanism in Internet of Things" *Processes*, vol. 10, no. 12, pp. 2631, 2022.
10. Anajemba .H, C. Iwendi, M. Mittal and T. Yue, "Improved Advance Encryption Standard with a Privacy Database Structure for IoT Nodes," *IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 201-206, 2020.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details