End-to-End Encryption in IoT System with AES Technique

A. V. Nageswara Rao, Bellamkonda Harshavardhan, Kollipara Sujith and Kyathi Priya
Department of Advanced Computer Science and Engineering, Vignan's Foundation for Science, Technology & Research (Deemed to be University), Vadlamudi, Andhra Pradesh, India

Keywords: End-to-End Encryption, Internet of Things (IoT), Advanced Encryption Standard (AES), Security, Privacy, Data Integrity, Cryptography.

Abstract: The Internet of Things (IoT) has deeply impacted various industries such as healthcare, smart home automation, transportation, and industrial automation. However, the rapid uptake of IoT technologies has introduced substantial concerns about security and privacy. Protecting the confidentiality and integrity of the data that is communicated between IoT devices, particularly as the devices are often found in exposed and vulnerable settings, is extremely important. End-to-end encryption (E2EE) is a reliable mechanism for encrypting data transfer between IoT devices so that the information can only be accessed by the intended sender and receiver. This article explores the use of end-to-end encryption in IoT systems using the Advanced Encryption Standard (AES) algorithm. The AES encryption technique achieves an optimum balance between computational cost and protection, making it highly appropriate for resource-constrained IoT devices. We discuss the structure of IoT systems, the security challenges they pose, and the incorporation of AES to enhance privacy and data integrity. Additionally, we evaluate the performance of AES encryption in IoT settings, considering aspects such as power usage, processing overhead, and scalability.

1 INTRODUCTION

The Internet of Things (IoT) covers a group of connected devices that are able to communicate and exchange information via an autonomous action, without direct human involvement. This network involves sensors, actuators, and other common objects that have computing power and are the ability to join networks. IoT use is common in various industries, including smart homes, healthcare, smart cities, and industry automation.

With the increased spread of IoT devices, the protection of the information shared through these networks has become an important issue. The confidential nature of the data being shared, such as personal medical records and financial interactions, makes it critical to secure such data from misuse or unauthorized access. In such scenarios, end-to-end encryption (E2EE) has played a crucial role in the protection of data in IoT networks.

End-to-end encryption works by encrypting the data at the source (sender) and decrypting it exclusively at the destination (receiver) so that any middlemen or attackers cannot read the data as it is transmitted. The Advanced Encryption Standard (AES) is well recognized to be a safe and effective method of encryption and hence a perfect option for data protection in IoT settings. AES is a symmetric key block cipher that uses the same key for encryption and decryption, allowing for relatively quick and efficient processing—a critical consideration for IoT applications that seek to keep power consumption and computational requirements low. This paper discusses the use of AES-based end-to-end encryption in IoT systems, highlighting the benefits, challenges, and performance considerations involved in this approach.

IoT devices are constantly exchanging sensitive information, making them good targets for cyberattacks. Cybercriminals take advantage of vulnerabilities in communication protocols to intercept or tamper with data, leading to breaches and unauthorized access. The problem is how to deploy encryption mechanisms that provide strong security while maintaining computational efficiency. Given the resource constraints of many IoT devices. This paper presents an approach using AES encryption over TCP to enhance security without significantly impacting system performance.

Rao, A. V. N., Harshavardhan, B., Sujith, K. and Priya, K.
End-to-End Encryption in IoT System with AES Technique.
DOI: 10.5220/0013892800004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 3*, pages 105-112
ISBN: 978-989-758-777-1

1.1 Security Requirements

IoT systems are frequently deployed in environments that may be hostile or untrusted, leading to a heightened risk of cyberattacks. The primary security requirements for IoT systems include confidentiality, which ensures that only authorized parties can access the transmitted data; integrity, which guarantees that the data remains unaltered and untampered with during transmission; authentication, which verifies the identities of the devices involved in communication to prevent unauthorized access; non-repudiation, which ensures that the sender cannot deny having sent the message; and availability, which makes sure that the IoT system and its data are accessible to authorized users when needed.

IoT devices encounter a range of threats, such as eavesdropping, where unauthorized parties intercept data during transmission; Man-in-the-Middle (MitM) attacks, which involve interception and alteration of data between the sender and receiver; Denial-of-Service (DoS) attacks, which overload IoT devices or networks to render them unavailable; and physical attacks, where unauthorized access to IoT devices is used to extract sensitive information or compromise the system.

End-to-End Encryption (E2EE) protects IoT data by ensuring that only the intended recipient can read it. Even if hackers intercept the data, they will only see scrambled, unreadable text. Only the recipient with the correct decryption key can unlock and view the original information. To achieve this, different encryption techniques are used. For IoT systems, symmetric encryption, such as AES, is a popular choice because it is both fast and efficient, making it ideal for resource-limited devices.

1.2 AES Encryption Technique

The Advanced Encryption Standard (AES) is a symmetric key block cipher that encrypts data in fixed-size blocks of 128 bits. It supports three key sizes: 128 bits, 192 bits, and 256 bits. AES is widely recognized as a secure and efficient encryption algorithm, having been adopted by various organizations and standards, including the U.S. National Institute of Standards and Technology (NIST). AES functions through a series of well-defined rounds that include substitution, permutation, and mixing, which help it resist common cryptographic attacks. The strength of AES lies in its large key size (up to 256 bits), making it computationally impractical to break through brute force methods (S. P. Suresh, A. Kumar, and R. K. Sharma., 2020).

1.3 AES in IoT Systems

IoT devices often have limited resources, including processing power, memory, and battery life. Therefore, selecting an encryption algorithm that balances security with computational efficiency is crucial. AES is particularly well-suited for IoT systems for several reasons. It is highly efficient in both encryption and decryption, which is vital for IoT devices with limited processing capabilities. Additionally, AES has a lower computational overhead compared to other encryption algorithms, helping to reduce the energy usage of IoT devices. It is also highly scalable, making it suitable for both small and large-scale IoT networks. Furthermore, (N. Elgendy et al., 2023) AES offers robust encryption, making it resistant to brute-force attacks and other cryptographic vulnerabilities (] H. Zhang, Y. Liu, and X. Wang., 2023).

An IoT system generally includes several components. Devices gather data and transmit it over the network. The gateway or router serves as a bridge between IoT devices and cloud services or other devices. The cloud or server processes and stores the data collected by IoT devices, often providing additional functionalities.

1.4 IoT System Components

An IoT system generally includes the following components: Devices/Sensors, which are IoT devices that gather data and transmit it over the network; Gateway/Router, which serves as a bridge between IoT devices and cloud services or other devices; and Cloud/Server, which processes and stores the data collected by IoT devices, often providing additional functionalities (J. T. Olsson and P. Andersson., 2023).

2 LITERATURE REVIEW

End-to-end encryption (E2EE) is a security method that keeps data safe and unchanged while it moves from one device to another in an IoT system. This means that only the intended recipient can access the original information, keeping it safe from hackers or unauthorized access. One of the most commonly used encryption methods for this is AES (Advanced Encryption Standard). AES is a type of encryption that uses the same key to both lock (encrypt) and unlock (decrypt) the data. It is popular in IoT because

it provides strong security without slowing down devices, which often have limited processing power. This survey looks at how AES is used in IoT to protect data, discussing different ways it has been implemented, the frameworks that support it, and the challenges researchers have found in making it even better.

2.1 ISA 100.11a

Several IoT frameworks have integrated AES to secure data transmission effectively. The ISA 100.11a standard uses AES-128 for data confidentiality and message integrity. This standard employer's device authentication and freshness checks to prevent replay attacks. These mechanisms ensure that IoT systems maintain a high level of data security during trans mission and processing.

2.2 6LoWPAN

Another key approach is 6LoWPAN, which combines IEEE 802.15.4 with IPv6. This framework uses AES-based security modes at the link layer and integrates IPsec at the network layer, offering comprehensive end-to-end encryption. This dual-layer encryption enhances both the security and reliability of data exchanges in low-power, low-bandwidth IoT environments.

2.3 LoRa Wan

LoRa WAN implements a dual-layer encryption mechanism based on AES. It uses a network session key to protect communication between devices and network servers and an application session key to ensure encryption at the application level. This layered security architecture enhances the secure transmissions of data across long-range IoT networks.

2.4 Adaptive Framework

Given the limited computational resources of many IoT devices, adaptive and lightweight implementations of AES have been proposed. One such approach involves an adaptive framework that considers five different AES implementation schemes. This framework uses the Hungarian algorithm to optimize resource consumption and throughput, balancing security and performance in heterogeneous IoT environments

2.5 Lightweight Encryptography

NIST's lightweight encryption competition has introduced encryption algorithms like GIFT-COFB and Tiny JAMBU, which provide authenticated encryption while maintaining efficiency. These lightweight algorithms are specifically designed to address the performance constraints of IoT devices while ensuring robust data protection.

2.6 AES and Blockchain

Hybrid encryption approaches combining AES with other cryptographic techniques offer enhanced security and flexi bility. One innovative framework integrates AES encryption with blockchain technology.

Table 1: Literature Review.

Study	Focus Area	Key Findings
ISA 100.11a	AES-128 for data confidentiality and message integrity.	Prevents replay attacks using linchpins for device authentication and freshness checks.
6LoWPAN	Combination of IEEE 802.15.4 with IPv6 and AES-based security modes.	Provides link-layer and network-layer security using Ipsec for comprehensive encryption.
LoRaWAN	Dual-layer AES encryption: network session key and application session key.	Ensures secure communication between end devices and network servers.
Adaptive Framework	Five AES implementation schemes optimized using the Hungarian algorithm.	Balances resource consumption and encryption throughput in heterogeneous IoT environments.

Lightweight Cryptography	NIST lightweight encryption competition, including GIFT-COFB and Tiny JAMBU	Offers authenticated encryption suitable for resource-constrained IoT devices
AES and Blockchain	AES-CBC mode encryption with dynamic private keys and smart contracts.	Ensures confidentiality, immutability, and traceability in IoT data communication.

IoT data is encrypted using AES in Cipher Block Chaining (CBC) mode with dynamically generated private keys, and the encrypted data is stored on Solidity-based smart contracts. This method ensures data con f identiality, integrity, and immutability while leveraging the security benefits of blockchain.

In conclusion, AES is a robust solution for securing data transmission in IoT systems through end-to-end encryption. Ongoing research focuses on improving key management strategies, optimizing encryption algorithms, and exploring hybrid encryption techniques to face the unique number of challenges posed by IoT ecosystems. Table 1 shows Literature Review.

3 EXISTING SOLUTIONS

End-to-End Encryption (E2EE) is increasingly being implemented in IoT systems to ensure secure data transmission between devices, gateways, and cloud services. Given the growing cybersecurity threats, E2EE is essential for protecting sensitive data from unauthorized access and tampering. Include Smart Homes Consumer IoT, where devices like smart cameras, door locks, and voice assistants use E2EE to secure communication between users and cloud services, often employing protocols like MQTT with TLS and HTTPS to encrypt data. In Healthcare IoT (IoMT), E2EE protects sensitive medical data transmitted between wearable health devices and healthcare providers, with standards like HIPAA enforcing encryption for remote patient monitoring.

Industrial IoT (IIoT) utilizes E2EE in manufacturing plants to protect real-time sensor data and control signals from cyber threats, with Secure SCADA (Supervisory Control and Data Acquisition) systems implementing AES encryption. Automotive IoT sees connected vehicles employing E2EE to protect vehicle-to-cloud and vehicle-to-vehicle (V2V) communications, and technologies like V2X (Vehicle-to-Everything) incorporate encryption to

prevent cyberattacks on autonomous systems. Smart Cities Infrastructure, encompassing IoT-enabled traffic systems, energy grids, and surveillance networks, uses E2EE to prevent hacking and unauthorized data access, with protocols like LoRaWAN and NB-IoT supporting encryption for securing large-scale deployments. However, current E2EE implementations face challenges such as computational overhead, where IoT devices with limited resources struggle with processing-intensive encryption, and key management, where secure key distribution and storage remain major concerns.

4 PROPOSED SOLUTION

The current approach to implementing IoT systems with AES follows these key steps:

- **IoT Devices**: Sensors and actuators collect data from their environment.
- **IoT Gateway**: Serves as a bridge, connecting devices to the cloud for secure data transfer.
- **Cloud Server**: Stores and processes the encrypted data received from IoT devices.
- **End-User Applications:** Provide users with access to the decrypted data through apps or dashboards.

4.1 Encryption and Transmission Process

Step 1: Key Generation and Distribution AES keys (128- bit, 192-bit, or 256-bit) are generated using a secure key management system.
Step 2: Data Encryption at the IoT Device. Data is encrypted using AES before transmission. Encrypted data is transmitted securely over MQTT or HTTPS.
Step 3: Secure Transmission via IoT Gateway. The gateway ensures secure routing but does not decrypt the data. TLS/DTLS is applied to protect data integrity during transmission.
Step 4: Decryption at the Cloud or End-User Application. The recipient decrypts data using the pre-shared AES key. The original plaintext data is recovered.

Step 5: Key Management and Security Enhancements. Periodic key rotation and multi-factor authentication (MFA) improve security.
Requirement Analysis: This phase focuses on identifying security threats in IoT systems, choosing suitable data transmission protocols, and understanding the hardware and software limitations of the system. The main objective is to establish the necessary security measures and assess the computational feasibility for IoT devices.

System Design: During this stage, the overall architecture of the encryption system is outlined. The AES encryption algorithm is chosen for its effective balance of security and efficiency. Communication protocols such as TCP and MQTT are incorporated into the system to guarantee secure data transmission. Furthermore, lightweight cryptographic techniques are evaluated to enhance performance on resource-limited IoT devices.

The implementation phase includes data collection, where IoT devices like Raspberry Pi and Arduino gather sensor data in real-time. The encryption process follows, where the gathered data is encrypted using the AES-128 encryption standard prior to transmission. The encryption key is securely exchanged between the sender and receiver to prevent unauthorized access.Data transmission is carried out using TCP/MQTT over a secure TLS/SSL layer, ensuring that the communication channel is safeguarded against cyber threats. Finally, decryption and processing take place, where the receiver uses the shared AES key to unlock the encrypted data and processes it for use within the IoT system. Figure 1 shows Block Diagram.

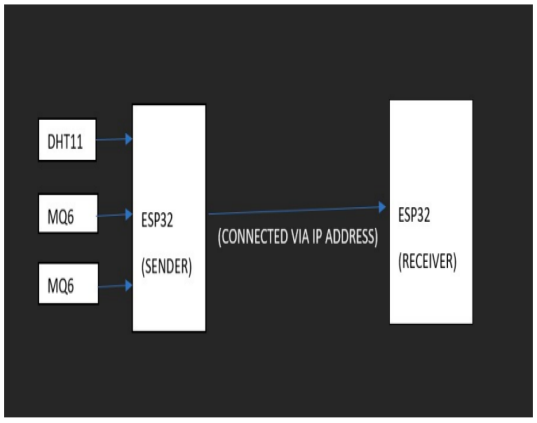

Figure 1: Block Diagram.

4.2 Technologies Used

Technologies Used: The implementation methodology adopts a systematic approach.
Requirement Analysis: This phase focuses on identifying security threats in IoT systems, choosing suitable data transmission protocols, and understanding the hardware and software limitations of the system. The main objective is to establish the necessary security measures and assess the computational feasibility for IoT devices.
System Design: During this stage, the overall architecture of the encryption system is outlined. The AES encryption algorithm is chosen for its effective balance of security and efficiency. Communication protocols such as TCP and MQTT are incorporated into the system to guarantee secure data transmission. Furthermore, lightweight cryptographic techniques are evaluated to enhance performance on resource-limited IoT devices.
The implementation phase includes data collection, where IoT devices like Raspberry Pi and Arduino gather sensor data in real-time. The encryption process follows, where the gathered data is m encrypted using the AES-128 encryption standard prior to transmission. The encryption key is securely exchanged between the sender and receiver to prevent unauthorized access.

Data transmission is carried out using TCP/MQTT over a secure TLS/SSL layer, ensuring that the communication channel is safeguarded against cyber threats. Finally, decryption and processing take place, where the receiver unlocks the encrypted data using the shared AES key and processes it for use within the IoT system. Figure 2 and 4 shows the flow diagram and figure 3 and 5 show the hardware connection.

4.3 Hardware Used

- **IoT Sensors Devices:** Collects real-time data (e.g., temperature, motion, humidity)
- **Dht11 sensors:** used for Temperature, Humidity
- **Gas sensors:** mq135, mq137
- **Microcontroller (Arduino/Raspberry Pi):** Processes sensor data and performs AES encryption.
- **Communication Module (Wi-Fi/Bluetooth/LoRa):** Transmits encrypted data over TCP/MQTT.

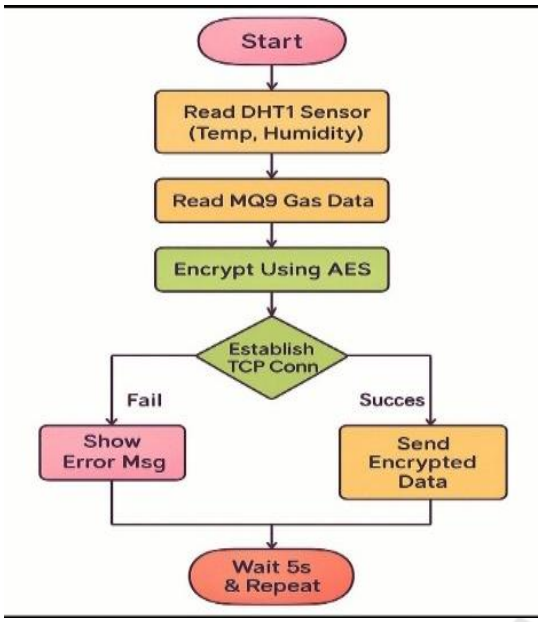

Figure 2: Sender Flowchart.

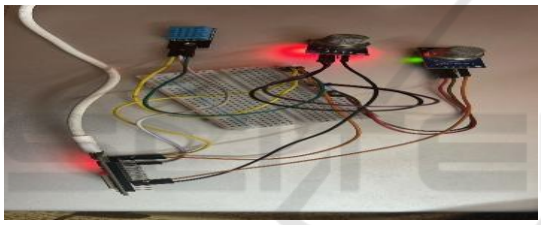

Figure 3: Physical Hardware.

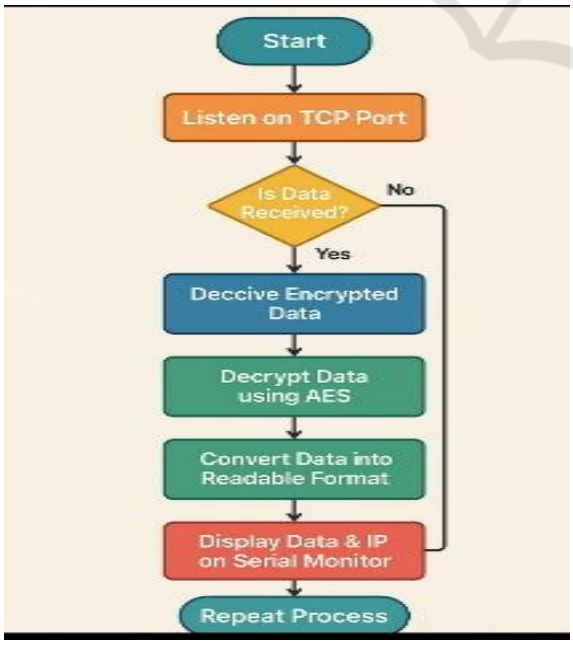

Figure 4: Receiver Flowchart.

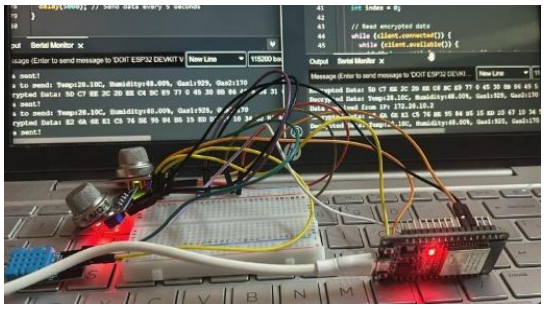

Figure 5: Device Connected to Laptop.

5 EXPERIMENTAL RESULTS

5.1 Sender Output

The ESP32 initiates sensor data collection by reading temperature and humidity from the DHT11 sensor connected to pin 4, as depicted in figure 4. It also acquires gas concentration values from two MQ6 gas sensors connected to pins 34 and 35. For network connectivity, the ESP32 establishes a connection to a WiFi network with the SSID "King kong" and the password "123456789". Subsequently, the ESP32 transmits the collected sensor data to a receiving device located at the IP address 172.20.10.5 and listening on port 12345. The data is sent repeatedly in the format "Temp:32.80C, Humidity:59.00", confirming continuous measurement and transmission. The serial monitor provides feedback by logging messages indicating successful data transmission, such as "Data to send: Temp:32.80C, Humidity:59.00Data sent!". This output confirms the ESP32's expected operation in acquiring sensor data and transmitting it over the WiFi network. Figure 6 shows the Sender.

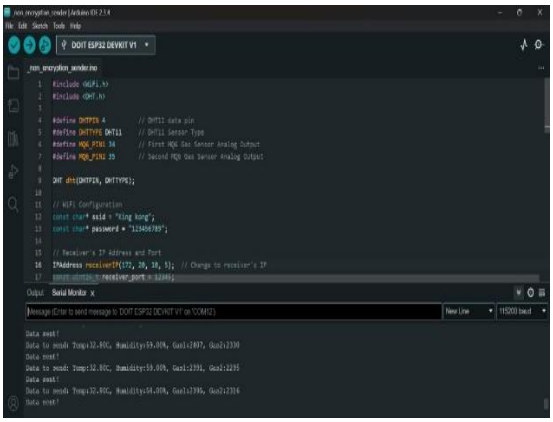

Figure 6: Sender.

5.2 Receiver Output

This output originates from an ESP32-based sender program designed to transmit temperature, humidity, and gas sensor data over WiFi to a designated receiver. The ESP32 begins by collecting sensor data, specifically temperature and humidity from a DHT11 sensor connected to pin 4, and gas concentration levels from two MQ6 gas sensors connected to pins 34 and 35. For network access, the ESP32 connects to a WiFi network identified by the SSID "King kong" and the password "123456789". The collected data is then directed to a receiver with the IP Address 172.20.10.5 on Port 12345. The transmission process involves repeatedly sending data in the format "Data to send: Temp:32.80C, Humidity:59.00Data sent!". The serial monitor serves as a confirmation mechanism, indicating successful transmission after each data packet is sent. Figure 7 shows the Receiver.

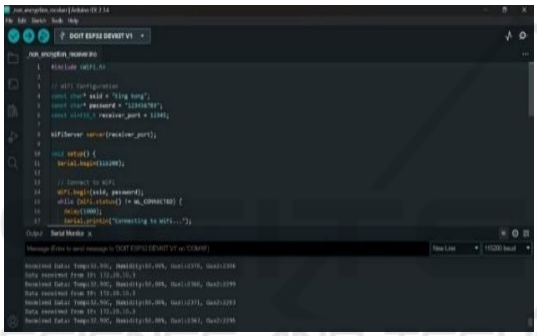

Figure 7: Receiver.

5.3 Encrypted Sender Output

The ESP32 initiates encrypted sensor data collection by reading temperature and humidity from a DHT11 sensor and gas concentration levels from two MQ6 gas sensors. For network connectivity, the ESP32 connects to a WiFi network with the SSID "King Kong" and the password "123456789". It then prepares to send encrypted data to a receiver at IP Address 172.20.10.5 and Port 12345. Prior to transmission, the sender encrypts the sensor data using AES with a specified AES Key (0x2B, 0x7E, 0x15, 0x16, 0x28, 0xD2, 0xa6, 0xab, 0xf7, 0x09, 0xcF, 0x4F, 0x3C, 0x76, 0x2F) and AES Initialization Vector (IV) (0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0C, 0x0C, 0x0D, 0x0D, 0x0D, 0x0). The decrypted data at the sender matches the original sensor reading (Temporary: 33.00C, Humidity: 58.00). The encrypted data itself appears as a long string of hexadecimal values (e.g., Ae Bf 19 62 C0 8E B5A ... 4A D6). The encrypted data is then sent to the receiver, which is expected to use the same AES key and IV for decryption. The serial monitor confirms successful transmission with the message "Data sent!". A key observation is that the sender successfully attaches and transmits data, and the encrypted data differs with each transmission due to the AES encryption process. The receiver is expected to decrypt and display the original sensor readings. Figure 8 shows the Encrypted sender.

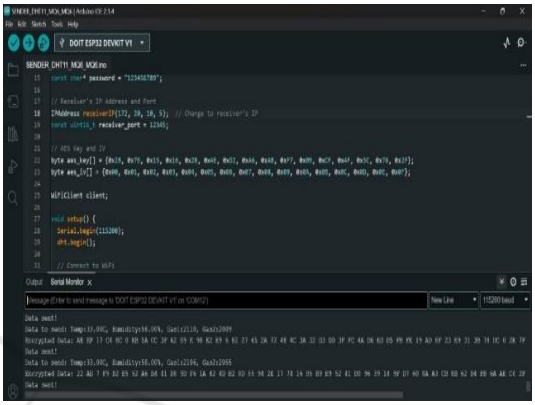

Figure 8: Encrypted Sender.

5.4 Encrypted Receiver Output

The ESP32, acting as an encrypted data receiver, connects to a WiFi network named "King Kang" and listens on Port 12345. It successfully receives encrypted data packets transmitted from another ESP32 sender over the WiFi network. Upon receiving data, the ESP32 initiates the encryption and decryption process. The received data, which is in AES-encrypted form, is initially displayed as a hexadecimal value (e.g., 22 AB 7F 29 ...). The ESP32 then decrypts this data using the pre-configured AES Key and IV, subsequently printing the decrypted sensor readings to the serial monitor. The serial monitor output displays "Decrypted sensor data: Decripted Data: Temp: 33.00C, Humidity: 58.00Encrypted Data: 22 AB7F 29 33 0 C 919D ... Get IP Address Data received from IP: 172.20.10.3". These decrypted values accurately represent the temperature, humidity, and gas sensor readings originally captured by the sender. This output confirms that the receiver is successfully decrypting the data transmitted by the sender, effectively demonstrating end-to-end encrypted communication.

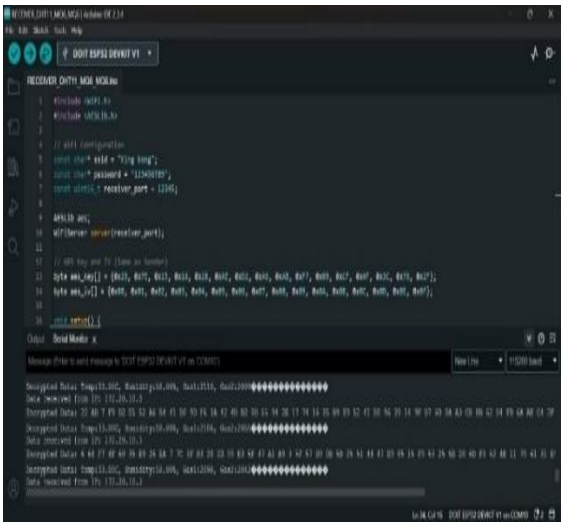

Figure 9: Encrypted Receiver.

6 CONCLUSION AND FUTURE WORK

This study shows that AES encryption is highly effective in protecting IoT systems from unauthorized access and cyber threats. It provides a strong yet efficient encryption method that works well even on devices with limited processing power. To ensure secure and reliable data transmission, the study also integrates TCP (Transmission Control Protocol), which guarantees that encrypted data is delivered in the correct order and without errors.

This makes AES and TCP a great com bination for IoT applications that require continuous, secure communication, such as smart homes, industrial automation, and healthcare monitoring. The results indicate that AES encryption adds very little delay while still maintaining high security, making it suitable for real-time applications. When compared to older encryption methods, AES proves to be more secure and more efficient, making it a preferred choice for modern IoT systems. Future Research Directions: Looking ahead, researchers can work on: Optimizing encryption for ultra-low-power IoT devices to enhance efficiency. Exploring hybrid en cryption methods to improve security while conserving energy. Developing lightweight cryptographic techniques to strengthen security in constrained environments, making se cure IoT systems more widely accessible. Figure 9 shows the Encrypted Receiver.

REFERENCES

Alaba, F. A., Othman, M., Hashem, I. A. T., Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88, 10-28.

Gurung, R. B., Lin, Y. D., Huang, Y. C. (2019). Lightweight Intrusion Detection for IoT Systems Using Convolutional Neural Networks. IEEE Internet of Things Journal, 6(5), 8170-8179.Improving Chain of Custody in Forensic Investigation of Electronic Digital Systems Giuliano Giova.

H. Zhang, Y. Liu, and X. Wang, "Research on Data Encryption Standard Based on AES Algorithm in Internet of Things Environment," International Conference on Advanced Technologies for Communications (ATC), 2020. Available: ResearchGate.

H. Zhang, Y. Liu, and X. Wang, "Research on Data Encryption Standard Based on AES Algorithm in Internet of Things Environment," International Conference on Advanced Technologies for Communications (ATC), 2020. Available: ResearchGate.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B. (2019). A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access, 7, 82721-82743.

J. T. Olsson and P. Andersson, "Enhancing End-to-End Communication Security in IoT Devices Through Application Layer Protocol," Master's Thesis, Uppsala University, 2023. Available: DiVA Portal.

Liu, Z., Zhang, Y., Xu, L. (2018). Secure and efficient data transmission in IoT-enabled intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 19(7), 2174-2184.

M. I. Nofal, A. A. Ghoneim, and R. K. Iqbal, "A Novel Secure End-toEnd IoT Communication Scheme Using Lightweight Cryptography Based on Block Cipher," Applied Sciences, vol. 12, no. 17, p. 8817, 2022. DOI: 10.3390/app12178817.

N. Elgendy, O. Ismail, M. G. Alajmi, and A. M. Elhouni, "Hybrid Cryptographic End-to-End Encryption Method for Protecting IoT Devices Against MitM Attacks," International Journal of Computer Science and Information Security (IJCSIS), vol. 19, no. 3, pp. 112–121, 2023. Available: ResearchGate.

S. P. Suresh, A. Kumar, and R. K. Sharma, "Enhanced Encryption Technique for Secure IoT Data Transmission," International Journal of Engineering Research Technology (IJERT), vol. 9, no. 10, pp. 45–52, 2020. Available: Academia.edu.

Singh, P. K., Rajan, R., Gupta, B. B., Conti, M. (2020). Securing the Internet of Things: A survey on security challenges, attacks, and countermeasures. Internet of Things, 12, 100313.Providing Cryptographic security and evidentiary Chain-of-Custody with the advanced forensic format, library, and tools1 Simson L. Garfinkel, Naval Postgraduate School and Harvard University, USA.