International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Zigbee Based Secure Wireless Communication Using AES

N. Deepthi
UG Student
deepthibujji160@gmail.com

B. Pramodhini
UG Student
bandelapramodini@gmail.com

R. Veda sri
UG Student
orrepalavedasri@gmail.com

Ms. G. Swathi
Assistant Professor
Swathi.gangula@vmtw.in

**Department Of Electronics and Communication Engineering**
Vignan's Institute of Management and Technology for Women,
Kondapur(V), Ghatkesar (M), Medchal Dist -501301 Telangana State.

**ABSTRACT**

In the rapidly growing domain of wireless communication, data security has emerged as a critical requirement, especially in resource-constrained embedded systems. This paper proposes a secure wireless communication system based on ZigBee technology and Advanced Encryption Standard (AES), implemented using Arduino microcontrollers. The system comprises a transmitter that accepts data via a keyboard, encrypts it using AES-128, and transmits it wirelessly via ZigBee. The receiver decrypts the data and displays it on an LCD. Leveraging ZigBee's low-power and low-data-rate characteristics, the design ensures secure, efficient, and scalable communication suitable for applications such as home automation, military systems, and industrial control. The integration of AES encryption ensures end-to-end data confidentiality without significantly impacting transmission speed or energy consumption. Performance evaluation confirms the system's viability for Internet of Things (IoT) applications demanding both security and efficiency.

**Keywords**: Wireless Communication, ZigBee Protocol, AES-128 Encryption, Arduino Microcontroller, Secure Data Transmission, End-to-End Encryption.

Introduction:

With the advent of the Internet of Things (IoT) and the proliferation of smart devices, wireless communication systems have become increasingly prevalent. However, this growth has been accompanied by a surge in concerns regarding data security and privacy. Traditional wireless systems are often vulnerable to eavesdropping and unauthorized access due to the lack of robust encryption mechanisms. ZigBee, a wireless protocol based on the IEEE 802.15.4 standard, offers a compelling solution due to its low power consumption, simplicity, and support for mesh networking. This paper introduces a secure communication framework using ZigBee and AES-128 encryption, implemented via Arduino microcontrollers. The system is designed to ensure confidentiality and integrity of data in low-power embedded applications without compromising energy efficiency or processing performance. The transmitter unit captures user input, encrypts it using AES, and sends it through ZigBee, while the receiver decrypts the message and displays it on an LCD. This cost-effective design presents a secure and scalable solution suitable for diverse domains including home automation, industrial control, and defense communication systems.

**Objectives**

1. To develop a secure and energy-efficient wireless

communication system using ZigBee protocol.

2. To integrate AES-128 encryption for ensuring confidentiality and integrity of transmitted data.

3. To implement the system using Arduino microcontrollers for cost-effectiveness and scalability.

4. To evaluate the performance of the system in terms of data security, transmission speed, and energy efficiency.

5. To design a robust communication framework applicable to various IoT-based and real-time applications.

**Problem Statement:**

As wireless communication continues to evolve and expand, the need for secure data transmission becomes more pressing. Conventional wireless systems often lack sufficient encryption mechanisms, making them vulnerable to interception, unauthorized access, and data tampering. These security gaps are particularly concerning in low-power IoT applications, where traditional encryption methods may be too resource-intensive. There is a need for a lightweight, secure communication framework that ensures data confidentiality without compromising on speed, power consumption, or scalability.

## II. Literature Review:

Numerous studies have addressed security in wireless communications, particularly focusing on low-power sensor networks and IoT devices. ZigBee has emerged as a popular protocol due to its low energy usage, suitability for short-range communication, and support for mesh networking. Research by Akyildiz etal. (2002) emphasized the potential of ZigBee in sensor networks, though initial implementations lacked strong encryption protocols.

Later works, such as those by Perrig et al. (2004), explored the integration of lightweight cryptographic algorithms in resource-constrained environments, highlighting the trade-offs between security and performance. AES, particularly the AES- 128 variant, has gained traction due to its robustness and moderate computational requirements, making it suitable for embedded systems.

Recent implementations using Arduino and ZigBee modules have shown promise in various domains, including smart homes and industrial automation. However, many of these applications either ignore encryption or implement custom, less secure algorithms. This paper builds upon these foundations by combining the security of AES with the energy efficiency of ZigBee, providing a practical and secure solution for modern wireless communication challenges.

## III. System Architecture and Components Block Diagram:

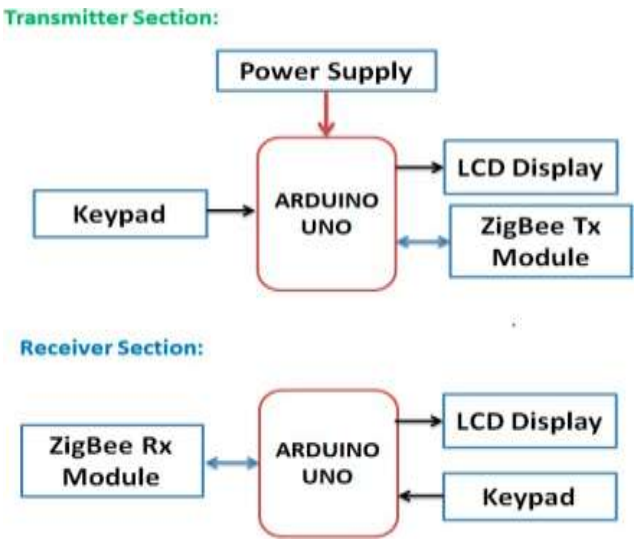


Fig 1. System Block Diagram

**Hardware Components:**

**1. Arduino uno**: The Arduino Uno acts as the central processing unit in both the transmitter and

receiver sections. In the transmitter, it takes user input from the keypad, encrypts the data using AES-128, and sends it to the ZigBee transmitter module. In the receiver, it receives the encrypted data from the ZigBee receiver module, decrypts it using the same AES algorithm, and displays the result on the LCD. It also handles communication with the keypad and LCD for user interaction


Fig 2. Arduino uno

2. **Keypad:** The keypad serves as the input interface for the user to enter alphanumeric data. Typicallyconfigured as a 4x4 matrix, it allows users to compose messages or commands, which are then processed by the Arduino. An optional keypad on the receiver side can be used for replies, commands, or confirmation inputs.

Fig 3. Keypad

**3. ZigBee TX module:** The ZigBee transmitter module is responsible for sending encrypted data wirelessly from the Arduino to a remote receiver. Operating under the IEEE 802.15.4 standard, ZigBee offers low-power, short-range, and reliable communication, making it ideal for secure data transmission in embedded systems.


Fig 4. ZigBee TX module

**4. ZigBee RX module:**

The ZigBee receiver module receives encrypted data wirelessly from the paired transmitter module. It communicates with the Arduino Uno over serial interface and forwards the received ciphertext to be decrypted and displayed. ZigBee ensures integrity and low-latency delivery of data even in noisy environments.


Fig 5. ZigBee RX module

**5. LCD Display:**

☞ **Wire (I2C) Library:** Manages the I2C protocol to interface with the LCD display, allowing real-time display of sensor readings and system status.

## IV. Working Principle

The system operates in two parts: the The LCD display (typically 16x2 characters) provides visual feedback to users. In the transmitter section, it shows the typed message and status updates such as encryption and transmission confirmation. In the receiver section, it displays the decrypted message, making the system easy to interact with and debug.


Fig 6: LCD Display

**Software and Programming**

The system is programmed using:

☞ **Arduino IDE:** Used for writing, compiling, and uploading the firmware to the ESP32 microcontroller. It provides an easy-to-use environment for developing embedded applications.

☞ **Embedded C/C++ Programming:** Implements sensor data acquisition from DHT11 and gas sensors, controls the DC fan, manages data logging on the SDtransmitter and the receiver. In the transmitter section, the user inputs a message through the keypad, which is then read by the Arduino Uno microcontroller. The input message is encrypted using the AES-128 algorithm to ensure data confidentiality during transmission. After encryption, the secured data is sent to the ZigBee transmitter module via serial communication (UART), which wirelessly transmits the encrypted message.

On the receiver side, the ZigBee receiver module receives the encrypted data and sends it to the second Arduino Uno. This Arduino decrypts the data using the same AES-128 algorithm and displays the original, readable message on the connected LCD display. This end-to-end communication ensures that the message is protected from unauthorized access during transmission, providing a secure and efficient wireless communication system. The optional keypad on the receiver side can be used for acknowledgments or further interaction.

## V. RESULTS

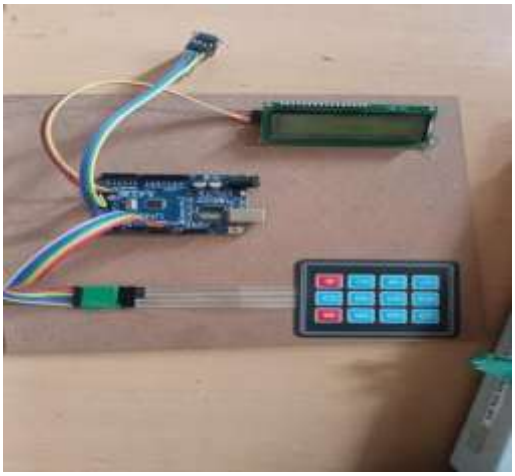

Fig 7. ZigBee TX module


Fig 8. ZigBee RX module

## VI. FUTURE ENHANCEMENTS

☞ **Bi-directional Communication**: The current system is unidirectional. Future versions can include two-way communication where both transmitter and receiver can send and receive encrypted data.

**Multiple Node Support**: Implementing ZigBee mesh networking can allow communication between multiple transmitter-receiver pairs in a distributed environment like smart homes or industrial setups.

**Cloud Integration**: Secure data logs can be uploaded to a cloud server for remote monitoring, control, and backup.

**Mobile App Interface**: A dedicated Android/iOS application can be developed to interact with the system via Bluetooth or Wi-Fi for remote access and control.

**Biometric Authentication**: Enhance system security by integrating biometric authentication (fingerprint or facial recognition) before sending sensitive data.

**Low-Power Optimization**: Power-saving techniques can be implemented for battery-operated deployments in IoT environments

## VII. CONCLUSION:

This project successfully demonstrates a secure wireless communication system using ZigBee technology and AES-128 encryption implemented on Arduino microcontrollers. By encrypting the data at the source and decrypting it only at the destination, the system ensures confidentiality and prevents unauthorized interception. The use of ZigBee makes the solution low-power and cost-effective, suitable for applications in home automation, military communication, and industrial control. The integration of AES encryption further strengthens the security, making the system reliable and scalable for future IoT applications. Overall, the proposed design offers a lightweight, secure, and efficient method of wireless data transmission in embedded environments.

## VIII. REFERENCES:

[1]. M. Zhang and Q. Liu, "Wireless Sensor Networks in Cold Chain Systems: A Review," *Journal of Industrial Information Integration*, vol. 18, p. 100147, 2020. [Online]. Available: https://doi.org/10.1016/j.jii.2020.100147

[2]. P. Sharma, S. Rathi, and K. Goyal, "Cloud-based Data Analytics for Cold Chain Monitoring," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 890–899, 2021. [Online]. Available: https://doi.org/10.1109/JIOT.2020.3021235

[3]. Y. Lee and S. Kim, "IoT-enabled Cold Storage Monitoring for Agriculture Using ESP32 and Firebase," *Proc. Int. Conf. on IoT in Agriculture (ICIA)*, Seoul, South Korea, 2022.

[4]. R. Dasgupta, A. Sengupta, and D. Banerjee, "Low-Cost Gas Detection Using MQ Sensors with ESP8266 for Cold Storage Applications," *Int. J. of Adv. Res. in Electronics and Communication Engineering*, vol. 9, no. 7, pp. 45–50, 2020.

[5]. J. Thomas and M. Rajan, "Edge AI for Cold Chain Management: Predictive Maintenance and Anomaly Detection," *Journal of Embedded Intelligence*, vol. 3, no. 1, pp. 1–12, 2023. [Online]. Available: https://doi.org/10.1016/jei.2023.01.001

[6]. V. Menon, "Data Visualization in Cold Chain Systems Using Cloud Dashboards," *Proc. Int. Conf. on Smart Logistics and SCM*, pp. 57–65, 2021.

[7]. Espressif Systems, "ESP32 Technical Reference Manual," 2023. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/

[8]. Arduino.cc, "ESP32 with Arduino IDE: Interfacing Sensors and SD Card," 2024. [Online]. Available: https://www.arduino.cc