International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 05 | May 2026 www.irjet.net p-ISSN: 2395-0072

Secure Data Communication in IoT using Cryptography

Mrs. Sushma P S[1], Sumanth B[2], Preethi K M[3], Sinchana V[4], Yashvanth H T[5]

[1]*Assistant Professor, Department of ECE. PES College of Engineering Mandya, Karantaka, India*
[2345]*Students of ECE, PES College Of Engineering Mandya, Karnataka, India*

Abstract - *The increasing adoption of the Internet of Things (IoT) demands robust yet efficient security mechanisms to protect sensitive data. Conventional software-based cryptographic techniques are often constrained by the limited processing power, memory, and energy of IoT devices. By utilizing a Secure Element as a hardware root of trust to provide non-exportable storage for a Master Secret (Km), this project implements a hybrid software-hardware cryptographic architecture. Instead of exposing long-term secrets, the system derives unique, ephemeral session keys (Ks) for every transmission, effectively eliminating static key exposure in host memory. This is achieved while leveraging dedicated hardware-accelerated AES-128 encryption on the ESP32 combined with software-based GCM authentication. This approach ensures strong data encryption while optimizing speed, energy consumption, and resource utilization.*

Key Words: **ESP32, Cryptography, ATECC608A, AES-128, GCM, Hardware Root of Trust.**

1. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has interconnected billions of devices across domains such as healthcare, smart homes, transportation, and industrial automation, enabling efficient data exchange but also creating major security and privacy challenges. Since IoT devices have limited processing power, memory, and energy, implementing strong yet lightweight security mechanisms is crucial. One of the biggest concerns is the risk of unauthorized access, data interception, and manipulation, which makes cryptography essential to ensure confidentiality, integrity, and authenticity of transmitted data. However, conventional cryptographic techniques are often too resource-intensive, highlighting the need for optimized algorithms designed specifically for low- power embedded environments. This project focuses on designing and integrating a hardware-backed lightweight cryptographic solution for IoT data protection.

1.1 LITERATURE SURVEY

Recent studies highlight that while software-based lightweight encryption reduces resource consumption, it often compromises security margins and introduces unacceptable CPU latency in constrained IoT devices. Furthermore, storing static cryptographic keys in standard microcontroller flash memory exposes them to physical extraction via JTAG/SWD debug ports, a critical vulnerability that hardware accelerators alone cannot resolve. Additionally, transmitting master keys over internal buses during provisioning creates network interception risks. To address these gaps, current research emphasizes combining Authenticated Encryption with Associated Data (AEAD) with an external Hardware Root of Trust (RoT). This project implements this exact architecture by adopting a "Zero-Key-in-Flash" policy. By utilizing the ATECC608A to permanently isolate the master secret and derive unique ephemeral session keys for the ESP32's AES engine, the system eliminates static key exposure, achieving high-speed, tamper-proof IoT communication.

2. PROPOSED METHODOLOGY

The proposed methodology introduces a hybrid cryptographic architecture that fundamentally mitigates standard microcontroller vulnerabilities through a strict "Zero-Key-in- Flash" policy. By permanently isolating the Master Secret within the non-exportable EEPROM of the Microchip ATECC608A secure element, the system neutralizes risks associated with physical memory extraction. At the edge node, an ESP32 microcontroller acquires real-time environmental telemetry via a DHT11 sensor. To secure this plaintext data without inducing processing latency, the architecture leverages the ESP32's native hardware accelerator for AES-128 encryption, complemented by a software implemented Galois/Counter Mode (GCM) to ensure robust data authentication. The cryptographic process generates a structured binary payload comprising the ciphertext, a unique Initialization Vector (IV), and a Message Authentication Code (MAC) tag, which is subsequently transmitted to a backend server over Wi-Fi utilizing the low-overhead User Datagram Protocol (UDP).

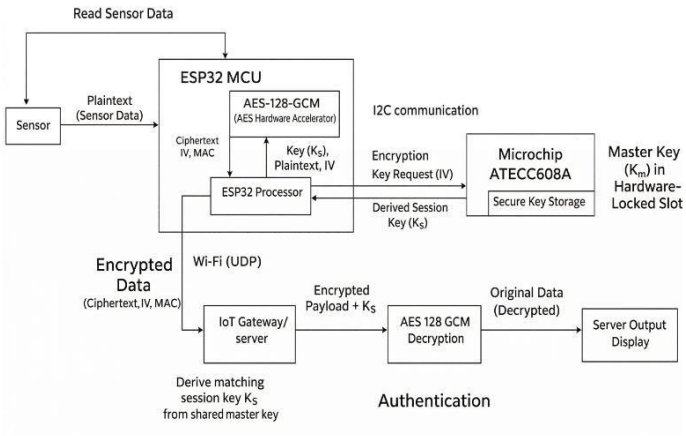

Fig1: Block diagram of Secure Data Communication in IoT using Cryptography

3. WORKING MECHANISM

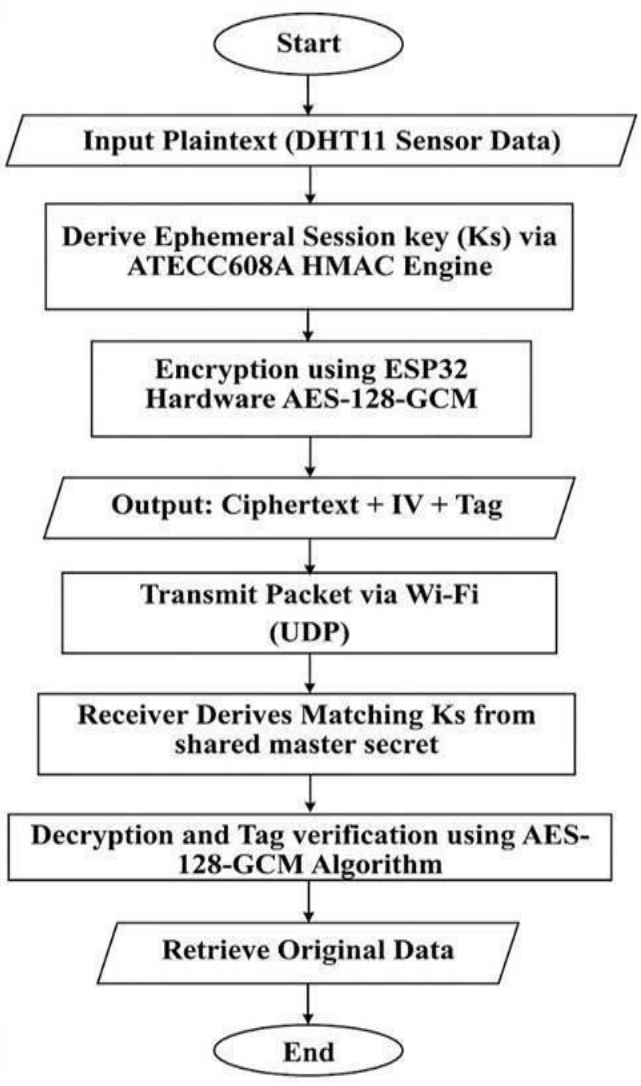

Fig 2: Flow chart for Secure Data Communication in IoT using Cryptography.

The operational workflow of the proposed hybrid secured data communication architecture is designed to establish an impenetrable, end-to-end authenticated encryption pipeline for IoT telemetry. The process initiates at the IoT Edge Node, where the ESP32 microcontroller systematically requests and acquires real-time plaintext environmental telemetry— specifically temperature and humidity—from the interfaced DHT11 sensor.

To establish a shared symmetric secret without exposing it to network interception during transmission, the system relies on an isolated, one-time secure provisioning phase. The ESP32 utilizes its internal True Random Number Generator (TRNG) to generate a high-entropy Master Secret (Km). This secret is injected directly into the dedicated Secure EEPROM slots of the Microchip ATECC608A cryptographic co-processor. Crucially, once injected, this slot is permanently hardware- locked, rendering the Master Secret completely non-exportable and invisible to the ESP32's external software memory.

During the active runtime phase, the system enforces Perfect Forward Secrecy by never utilizing the static Km for direct data encryption. Instead, for every transmission cycle, the ESP32 initiates a key derivation request over the I2C bus. The ATECC608A utilizes its internal HMAC engine, combining the locked Master Secret with cryptographic nonces to compute a unique 128-bit ephemeral Session Key (Ks).

This ephemeral Ks is temporarily loaded into the ESP32's hardware accelerator registers. The system then employs a highly optimized hybrid cryptographic approach: it leverages the ESP32's dedicated silicon to perform high-speed AES-128 encryption on the sensor data, while simultaneously utilizing a software-based implementation to compute the Galois/Counter Mode (GCM) authentication. This dual-layer approach outputs a confidential ciphertext and a highly secure Message Authentication Code (MAC) tag.

The ESP32 concatenates these cryptographic outputs into a structured binary frame following a strict byte-offset protocol: a 32-byte Initialization Vector (IV) header, a 16-byte GCM MAC tag, and the encrypted ciphertext payload. To maximize real-time telemetry throughput and minimize communication latency, this binary payload is transmitted to the backend server via Wi-Fi utilizing the User Datagram Protocol (UDP). Upon reception, the Python-based server independently derives the exact matching session key using its own secure copy of the Master Secret. It recomputes the MAC over the received ciphertext and IV; if the tags match, data integrity and authenticity are mathematically proven, and the payload is decrypted. Packets failing verification are automatically rejected to prevent tampering and ensure integrity.

4. DESIGN AND IMPLEMENTATION

The project implements a secure IoT communication pipeline by integrating the ESP32 microcontroller with the ATECC608A hardware secure element and a Python-based backend server. The system begins with reliable I2C communication setup, including wake-up signaling and diagnostic verification of the secure chip. A critical provisioning phase ensures strong security by locking configuration zones, generating cryptographic keys using the ESP32's TRNG, and securely injecting a master key into the hardware. During runtime, sensor data from the DHT11 is collected and protected using dynamically derived session keys (via HMAC-SHA256), ensuring Perfect Forward Secrecy. The data is encrypted using AES-

128 GCM with hardware acceleration, producing ciphertext and an authentication tag, then transmitted over Wi-Fi using UDP in a structured packet format.

On the backend, a Python server listens on UDP, parses incoming packets, and reconstructs the session key using the shared master key and received IV. It then performs AES-GCM decryption and integrity verification, rejecting any tampered data. Replay attacks are mitigated using a fast lookup mechanism that tracks previously used IVs. The design balances strong security with resource efficiency by offloading cryptographic operations to dedicated hardware, minimizing CPU and memory usage on the ESP32. Overall, the system ensures secure key storage, encrypted communication, data integrity, and resilience against common IoT attacks such as interception and replay.

5. CIRCUIT EXPLANATION

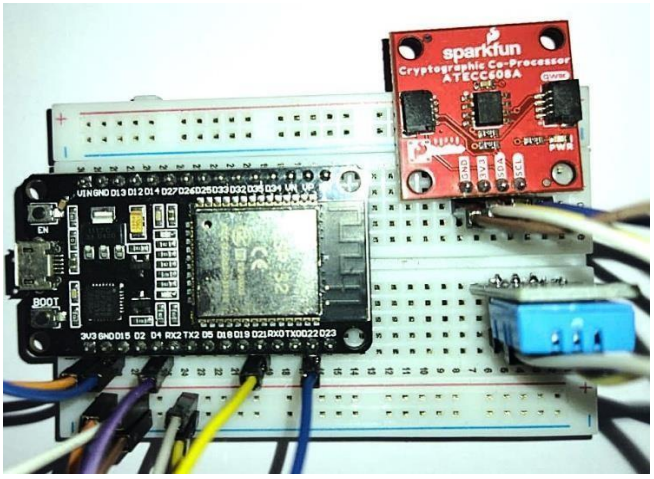

Fig 3: System circuit connection

The hardware architecture functions around the ESP32 development board, powered by a dual-core Tensilica LX6 processor, acting as the primary centralized microcontroller. A DHT11 sensor is interfaced via a single-wire digital GPIO pin to acquire temperature and humidity measurements. The cornerstone of the security architecture is the Microchip ATECC608A cryptographic co-processor, interfaced with the ESP32 via the standard I2C communication bus (SDA and SCL pins).

This secure element serves as the Hardware Root of Trust, isolating all sensitive key derivation operations away from the main MCU's vulnerable volatile memory. To guarantee absolute physical security, the configuration zone and data zone of the ATECC608A are permanently locked post-provisioning, a state securely verified by a 0x55 hex lock value in the configuration dump. This irreversible hardware lock ensures the embedded master key cannot be extracted, exported, or read by any external software command or physical debug-port probing.

To ensure reliable synchronous communication, the I^2C bus connecting the ESP32 and the ATECC608A is electrically stabilized using standard pull-up resistors on the Serial Data (SDA) and Serial Clock (SCL) lines. Both the primary microcontroller and the cryptographic co-processor operate efficiently within a unified 3.3V power domain. This shared logic level eliminates the need for intermediate voltage-level shifters, thereby reducing the overall circuit footprint and minimizing hardware latency. The DHT11 sensor, which also operates compatibly within this logic range, transmits its single-wire digital signal directly to an allocated ESP32 GPIO pin [4] for real- time edge processing.

6. RESULTS AND DISCUSSION

The empirical implementation of the architecture validates the feasibility and superiority of a hybrid hardware-software cryptographic model for constrained IoT edge devices. During sustained operational testing, the ESP32's hardware-accelerated AES-128 engine processed the payload encryption with negligible CPU stalling, allowing the Tensilica LX6 cores to maintain network stability. The software-based GCM implementation seamlessly appended the authentication tags without inducing the unacceptable processing latencies typically associated with full software-suite encryptions.

The Wi-Fi transmission of the serialized binary frames proved highly efficient under UDP. At the receiver backend, the Python environment accurately executed the decrypt and verify sequence. Simulated network attacks were conducted to rigorously evaluate the system's defensive mechanisms.

During simulated payload tampering—where single bits within the ciphertext or IV were intentionally flipped—the AES-GCM verification strictly failed. The server instantly triggered security alerts, recognizing the MAC mismatch, and actively dropped the corrupted data packets, confirming the system's absolute resilience against Man-in-the-Middle (MITM) data manipulation.

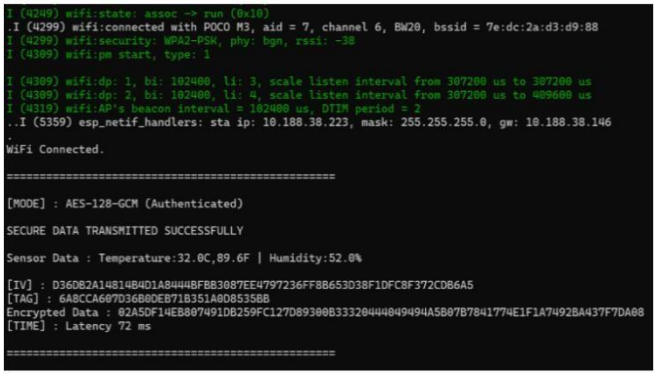

Fig4: Encrypted Data transmitting from esp32

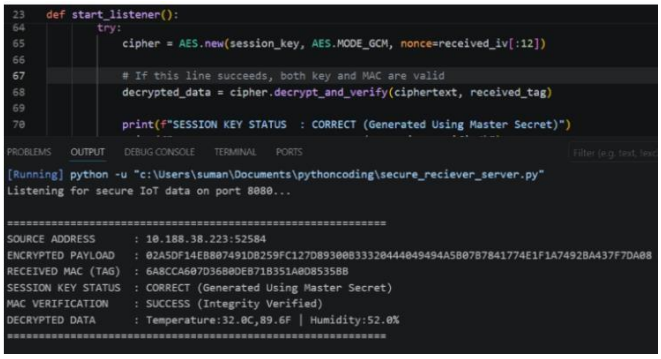

Fig5: successfully Received, Verified and Decrypted the IoT sensor data

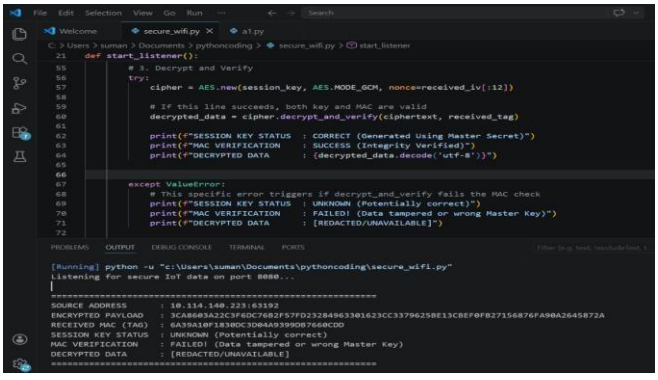

Fig6: Received Data Tampered Then data packet rejected

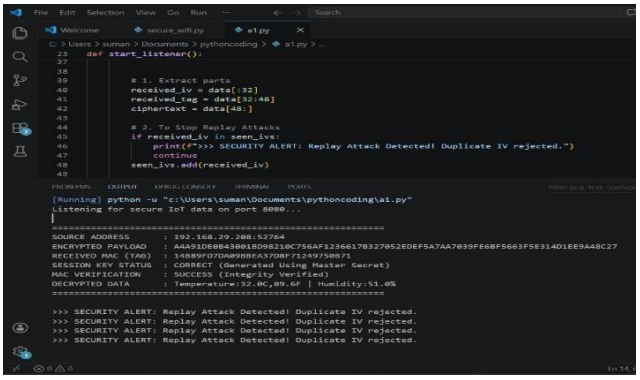

Fig7: Replay attack test result

7. OBSERVATION TABLE

Table 1 comprehensively outlines the system's empirical operational responses and cryptographic behaviors under both standardized functional telemetry conditions and simulated adversarial network attacks. During standard operation with valid DHT11 environmental data, the system successfully verifies the Message Authentication Code (MAC) and decrypts the payload, seamlessly ensuring data integrity and confidentiality. To rigorously validate the system's defensive mechanisms, various threat vectors were simulated; for instance, during a data tampering attack involving an intentionally altered ciphertext, the system immediately detected a MAC verification mismatch and forcefully rejected the corrupted packet. Furthermore, simulated physical memory attacks attempting to read the Master Key were successfully thwarted, as the key remained permanently isolated and inaccessible within the hardware-locked EEPROM of the ATECC608A secure element. Finally, network replay attacks were effectively neutralized by strictly enforcing Initialization Vector (IV) uniqueness, ensuring that the transmission of duplicate IVs resulted in immediate packet rejection, thereby proving the architecture's absolute resilience against sophisticated data manipulation and unauthorized access.

Table 1: Observation Table

SL.NO	Test Condition	Input Given	System Response
1	Standard Operation	Valid DHT11 Data	Integrity verified, Data Decrypted

2	Tampering Attack	Altered Ciphertext	MAC verification failed, Data rejected
3	Key Isolation	Master Key Read Attempt	Kept isolated inside ATECC608A
4	Replay Attack	Duplicate IV Transmitted	Attack prevented via unique IV

8. EXPECTED OUTCOMES

The primary objective of architecting a highly secure, tamper-proof IoT edge communication framework was definitively achieved. The strategic integration of the ATECC608A provides true Key-at-Rest security, enforcing a strict "Zero-Key-in-Flash" policy that fundamentally neutralizes the most critical vulnerability in modern microcontrollers: physical memory extraction.

Furthermore, the implementation of Ephemeral Session Isolation ensures that the foundational Master Secret is never loaded into the ESP32's volatile memory during active runtime encryption, vastly reducing the attack surface against buffer-overflow exploits. By intelligently distributing the cryptographic workload—utilizing hardware silicon for AES confidentiality and optimized software for GCM authenticity—the system achieves enterprise-grade data protection without sacrificing the energy efficiency or real-time processing capabilities required for IoT edge telemet.

9. FUTURE SCOPE

The established architecture provides a robust, scalable baseline for industrial telemetry networks. Because the framework utilizes a shared I2C bus for the secure element, it naturally supports the seamless future integration of high- precision, industrial-grade I2C sensors (such as the BMP280 or BME680). This expansion can be achieved without demanding complex firmware overhauls or interfering with the established hardware Root of Trust.

Future empirical research will focus on granular performance benchmarking. This includes precise oscilloscopic measurements of the AES-GCM hardware encryption latency, quantitative profiling of CPU clock cycle consumption, and exact RAM footprint tracking. Directly comparing these hybrid hardware-software metrics against traditional software-only lightweight cryptographic libraries will yield valuable data for optimizing future ultra-low-power industrial IoT architectures.

10. CONCLUSIONS

The proposed ESP32 and ATECC608A-backed cryptographic architecture effectively resolves the most critical vulnerabilities inherent in securing IoT sensor data networks. By delegating key management to a dedicated cryptographic co-processor acting as a Hardware Root of Trust, the system completely bypasses the severe risks associated with storing static keys in standard microcontroller flash memory. The dynamic computation of an ephemeral Session Key via the ATECC608A's HMAC engine ensures Perfect Forward Secrecy for every individual transmission cycle.

The implementation has proven highly resilient against diverse and sophisticated attack vectors, including eavesdropping, data modification, replay attacks, and physical memory dumping. Ultimately, this hybrid methodology delivers a robust, high- performance, and highly scalable foundation for securing data integrity and confidentiality in modern IoT deployments.

REFERENCES

1] S. H. Rafat, M. N. Jarin, T. M. Mahdee, et al., "Lightweight Cryptographic Algorithm Analysis for Secure IoT Communication on ESP-32 Platforms", IEEE International Conference on Quantum Photonics, Artificial Intelligence and Networking, 2025

2] Lei Zhao, Junhua Deng, Yimin Wang, Yulong Ma, Peng Lu, "Data Compression and Encryption Fusion: A Review of Hybrid Techniques for Secure and Efficient Online Transmission", IEEE Access, 2025

3] Siddharth Patel and Rohit Singh "Hardware-Enabled Root of Trust for Enhanced Security in Embedded Systerns", IEEE Xplore (13th IEEE International Conference on Intelligent Systems and Embedded Design),2025

4] Catarina Silva, Vitor A. Cunha, João P. Barraca, Rui L. Aguiar, "Analysis of the Cryptographic Algorithms in IoT Communications", Information Systems Frontiers (Springer), 2024

5] Ala Saleh D. Alluhaidan, P. Prabu, "End-to-End

Encryption in Resource-Constrained IoT Device". IEEE Access, 2023

6] Mohammad Al-Mashhadani, Mohamed Shujaa, "IoT Security using AES Encryption Technology Based ESP32 Platform", IAJIT Journal, 2022

7] Microchip Technology Inc., ATECC608A CryptoAuthentication Device Summary Datasheet, DS40001977B, 2022. [Online]. Available: https://www.microchip.com

8] Espressif Systems, ESP32 Technical Reference Manual, Version5.12024.
[Online].Available:https://www.espressif.com

9] Espressif Systems, ESP-IDF Programming Guide, Version5.1,2024.
[Online].Available:https://docs.espressif.com