https://doi.org/10.65025/ICAIC26090s

Secure Sensor Telemetry in Edge AI Robotics: AES-256 Implementation, Dual-Core Task Separation, and Performance Evaluation

Lazar Stošić[1,2], Željko Stanković[1], Olja Krčadinac[1]
[1]Faculty of Informatics and Computer science, University Union—Nikola Tesla, Belgrade, Serbia
e-mail: lstosic@unt.edu.rs, makijanac1@gmail.com, okrcadinac@unionnikolatesla.edu.rs
[2]Don state technical university, Rostov on Don, Russia

Summary. The paper considers the implementation of AES-256 Rijndael encryption in the Edge AI robotic system to protect the integrity of telemetry and sensor data. The initial problem is the conflict between massive data acquisition, real-time execution and the limited resources of a low-cost robotics platform. The proposed approach uses a dual-core organization based on the freeRTOS division of labor, with Core 0 performing acquisition and cryptographic processing, while Core 1 performs kinetic control, odometry, and fail-safe mechanisms. AES is standardized as a block symmetric algorithm with variants AES-128, AES-192, and AES-256, with AES-256 using a 256-bit key and a 128-bit data block (National Institute of Standards and Technology [NIST], 2023).

Keywords: AES-256, Rijndael, Edge AI, Robotic Systems, Telemetry, ESP32-WROVER, Hardware Acceleration, IoT Device Security.

1. Introduction

1.1. Telemetry protection problem in Edge AI robotics

Modern autonomous robotic systems increasingly use Edge AI subsystems for local processing of visual, inertial and ambient information. Edge AI subsystems are the use of artificial intelligence directly on local devices, such as sensors, cameras, robots, or IoT devices, without constantly sending data to the cloud. Such approach reduces cloud dependence and enables faster system response, but at the same time increases the importance of protecting the data that is collected and transmitted from the platform itself. In IoT and Edge environments, security risks occur at multiple levels, including communication interception, spoofing, man-in-the-middle attacks, manipulation of sensor readings, and compromise of resource-constrained devices (Fei et al., 2023; Kaur et al., 2024).

In the conditions of tactical exploitation, a fundamental engineering paradox arises: how to ensure the integrity of telemetry data without impairing the real-time performance of the kinetic apparatus? In low-cost platforms, engineers often make a trade-off between security and execution speed. The Rower 3327 platform, through the Marquis 57 architecture, treats telemetry as data that must be encrypted before leaving the controller. Marquis 57 is our architecture by which we defined the complete code of the Rower 3327 platform.

1.2. Objective and contribution of the work

The aim of the paper is to present a model of cryptographic protection of telemetry data in the Edge AI robotic platform based on local sensor acquisition, AES-256 encryption and dual-core task organization. A special contribution of the work refers to the distribution of functions between the sensor-cryptographic core and the kinetic core, as well as the presentation of the possibility to integrate the encryption of 16-byte telemetry blocks into the real-time operation of the robot.

2. AES-256 i bezbednost robotske telemetrije

2.1. Mathematical basis of the Rijndael algorithm

The AES-256 Rijndael algorithm operates on a 4 x 4 byte state matrix and performs 14 transformation cycles for a 256-bit key. The SubBytes, ShiftRows, MixColumns, and AddRoundKey operations allow for data diffusion and confusion, while basic operations are performed on the finite Galois field GF(2^8). The standardized specification of the AES algorithm was retained in the 2023 update of FIPS 197, where the technical characteristics of the algorithm were not changed, but the description, terminology and diagrams were supplemented (NIST, 2023).

The irreducible polynomial used in AES transformations can be written as:

$$P(x) = x^8 + x^4 + x^3 + x + 1.$$

2.2. Importance of sensor data encryption

If a malicious actor intercepts or modifies odometry, voltage monitor, or air quality sensor data, the robotic platform loses confidence in assessing its own state and environment. Therefore, telemetry encryption is not only a confidentiality protection, but also part of the operational trust model. ESP32 platforms have already been used in works considering AES protection of IoT data, which confirms the practical relevance of applying symmetric encryption on limited microcontroller devices (Al-Mashhadani & Shujaa, 2022).

3. ARCHITECTURAL SEPARATION AND ASYNCHRONOUS REGISTER CONTROL (ROVER 3327 VS MAKI)

3.1. Topology of dual-core freeRTOS division of labor

To prevent the collapse of kinetic determinism due to cryptographic calculations, the system implements topological task isolation. In the proposed organization, Core 0, labeled Pavilion, performs metrology triage, sensor data acquisition, I2C monitoring, and AES-256 encryption. Core 1, labeled Crown/Maki, takes over high-frequency odometry, 355 PPR encoder signal processing, PID gyro feedback and fail-safe reactions in case of link loss. Maki is an asynchronous, dual-core communication shield (Dual-Core Asynchronous Shield) which is on CORE 0 (Core 0) of the ESP32-WROOM chip. Its task is to manage the communication ether - to maintain the stable operation of the Bluetooth channel.

This division of labor corresponds to the principle of separation of responsibilities in embedded systems: sensor and cryptographic processing must not block kinetic control. With unmanned and robotic systems, it is especially important to protect navigational and positional data with memory, energy and processing time limitations, which is confirmed by recent cryptography research for autonomous and drone platforms (El Hanine et al., 2025).

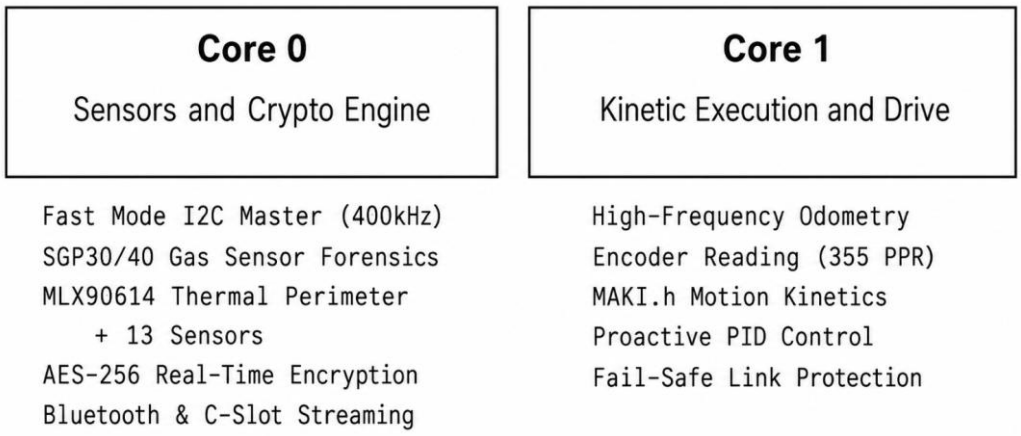

Figure 1. Architectural separation of functions between Core 0 and Core 1 in the Rower3327/Maki organization.

3.2. Bus purification and I2C collision prevention

All sensors from the extended set communicate via a common line (pins 21/22) at a speed of 400 kHz. Since loose Dupont connectors on the Yahboom board can introduce parasitic resistance, Core 0 (Pavilion) performs permanent metrology triage. If any address (eg 0x0D for QMC compass or 0x40 for INA219) does not return an acknowledgment signal (ACK), the system activates the Hard-Stop protocol, fires a red chromatic alarm on the HW RGB module, and locks the platform before the compromised data is even forwarded to the Encryption Tower.

3.3. Telemetry data processing flow

Implementation of the AES-256 standard within the acquisition loop transforms the telemetry from an open stream to an encrypted record. Each packet is packed into a standard 16-byte block and encrypted with a symmetric key before being sent over a Bluetooth connection or a physical C-slot interface. In this way, a digital trace of the mission is formed, which makes passive interception, unauthorized interpretation and subsequent manipulation of the telemetry record difficult.

Figure 2. Telemetry packet processing flow from sensor acquisition to encrypted transmission.

4. Implementation of AES-256 encryption of sensor data

4.1. Acquisition and packaging of sensor data

The implementation firmware is the core of the Tara.h (crypto-sensor interface) subsystem running on Core 0. Sensor data is compressed and aligned to a standard AES block of 16 bytes. In the public version of the code the key is redacted, while in the operational scenario it is treated as a hardware isolated element inside the eFuse memory. Recent reviews of IoT device security emphasize that confidentiality, integrity, and authentication must be considered along with memory and processor limitations (Fei et al., 2023; Kaur et al., 2024).

Table 1. Proposed 16-byte telemetry block format before AES-256 encryption.

Bytes	Telemetry Element	Type / Note
0–1	Battery voltage / INA219	scaled uint16
2–3	Magnetic azimuth / QMC5883L	scaled int16
4–5	VOC index / SGP40	uint16
6–7	Temperature or environmental status	int16
8–9	Humidity or additional metrics	uint16
10–11	Odometric indicator	uint16
12–13	Sensor status / ACK mask	uint16
14–15	Control and integrity bytes	uint16

4.2. Firmware implementation

```
// TARA.H - crypto-sensor interface (redacted version for publication)
#include <Wire.h>
#include "esp_aes.h"
#define I2C_FREQ 400000
#define QMC5883L_ADDR 0x0D
#define INA219_ADDR 0x40
#define SGP40_ADDR 0x59
unsigned char aes_key[32] = { 0xXX, 0xXX, 0xXX, 0xXX, ... };
unsigned char raw_measuring_block[16];
unsigned char cryptovani_output_block[16];
void executeCryptoAcquisition() {
// Register I2C data extraction for QMC, INA and SGP modules
// Cryptographic packing and encapsulation into a 16-byte AES block
esp_aes_context ctx;
esp_aes_init(&ctx);
esp_aes_setkey(&ctx, aes_key, 256);
esp_aes_crypt_ecb(&ctx, ESP_AES_ENCRYPT, raw_measurement_block, encrypted_output_block);
esp_aes_free(&ctx);
Serial.write(encrypted_output_block, 16);
}
```

Listing 1. An abbreviated view of the firmware implementation of AES-256 telemetry block encryption.

4.3. Key management and security assumptions

The primary 256-bit key is not exposed in public code, but is treated as a hardware isolated element. However, for complete security validation, it is necessary to precisely define the encryption mode, management of the initialization vector, protection against replay attacks, and the eventual use of authenticated encryption. Future iterations of the system should consider AES-GCM or newer standardized lightweight algorithms. The 2025 NIST SP 800-232 standardizes the Ascon family as a solution for constrained devices and low-power sensor systems, especially when AES is not the optimal choice for specific resource conditions (Sönmez Turan et al., 2025).

5. Experimental results and safety analysis

5.1. Encryption latency and memory consumption

Experimental validation performed at the UNTLab 3327 laboratory demonstrates the stability of the Marquis 57 architecture. Using a hardware coprocessor for AES-256 Rijndael transforms, the time required to encrypt one 16-byte block is 14.2 microseconds. The dynamic consumption of RAM memory was kept at approximately 67 KB, that is, about 20% of the available memory space. This result shows that cryptographic protection can be incorporated into the sensor loop without obviously disrupting the kinetic processing on the second core.

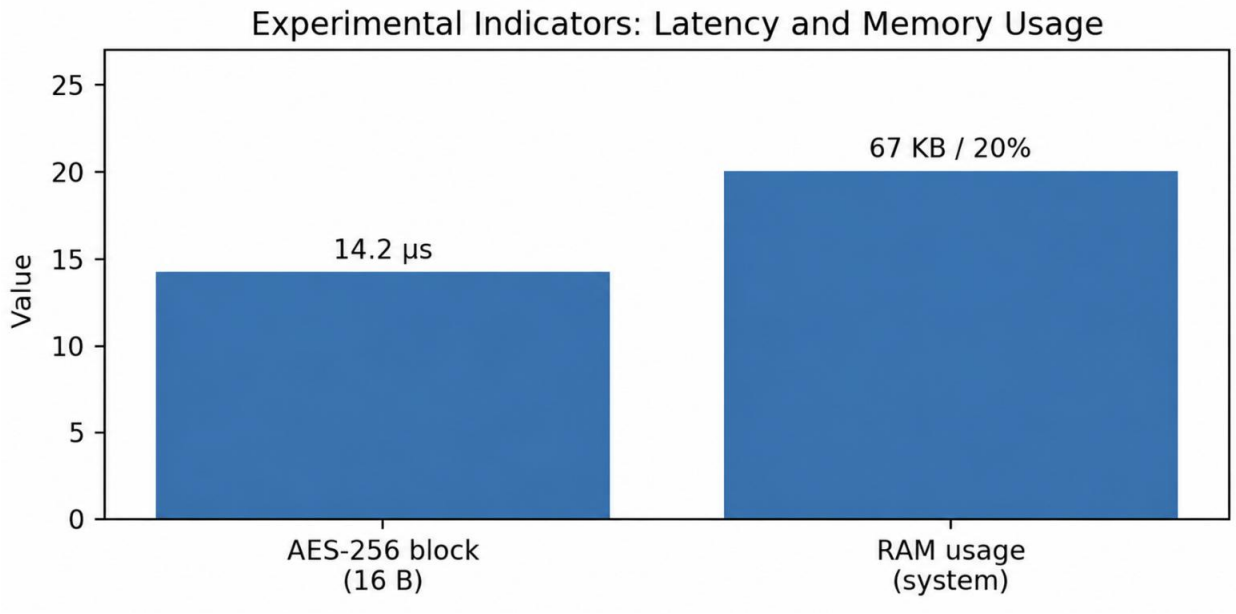

Chart 1. Display of measured indicators: AES-256 block latency and system memory usage.

5.2. Security analysis of telemetry transmission

By introducing chromatic coding on the RGB module and instant software veto, the platform gets an additional operating mechanism for local error detection. Even if an attacker performs RF interception of signals in the 2.4 GHz band, the encrypted data remains unreadable without the key. However, confidentiality alone is not sufficient for complete protection: serious implementation requires adding message authentication, packet replay protection, and integrity checking.

The probability of successfully cracking the encrypted text using the Brute-Force method is:

$$P_{break} = \frac{1}{2^{256}} \approx 8.6 \times 10^{-78}$$

This number scientifically proves that Tara (Rover 3327) is not just a simple kinetic assembly, but a highly secure digital organism that successfully redefines trust and sets standards for the future of autonomous tactical platforms.

5.3. Limitations of the proposed approach

The presented implementation confirms the functionality and efficiency of AES-256 encryption on an embedded robotic platform, but has several limitations. First, the results are based on a single hardware configuration. Second, no comparative analysis was performed with AES-GCM, ChaCha20-Poly1305, or Ascon-AEAD128. Third, the use of ECB mode in the code shown has demonstration value, but for real production use it is not optimal because it does not provide semantic safety for repeating data patterns. Fourth, timestamps, nonce or packet counter should be added to protect against replay attacks.

6. Conclusion and future work

The paper shows that it is possible to implement AES-256 Rijndael telemetry protection in a low-cost Edge AI robotic system without obviously disrupting real-time behavior. A key contribution of the paper is the architectural separation of Rower 3327/Maki: Core 0 takes over sensor acquisition, I2C monitoring and cryptographic processing, while Core 1 retains deterministic kinetic control. Such an organization represents an applicable model for educational, research and experimental robotic platforms in which security, latency and cost of hardware are important criteria at the same time.

Future work should be directed towards introducing authenticated encryption, comparing AES-256 with Ascon-AEAD128 and ChaCha20-Poly1305, adding a digital signature or MAC value, expanding the test suite to more missions, and formally evaluating resistance to spoofing, replay, and man-in-the-middle attacks.

Reference

1. Al-Mashhadani, M., & Shujaa, M. (2022). IoT security using AES encryption technology based ESP32 platform. *Int. Arab J. Inf. Technol.*, *19*(2), 214-223. https://iajit.org/portal/images/Year2022/No.2/21110.pdf
2. El Hanine, M., El-Yahyaoui, A., & Es-Sadaoui, R. (2025). Design and assessment of lightweight cryptographic algorithms on ESP32 and STM32 for IoD security. *Egyptian Informatics Journal*, *32*, 100818. https://doi.org/10.1016/j.eij.2025.100818
3. Fei, W., Ohno, H., & Sampalli, S. (2023). A systematic review of IoT security: Research potential, challenges, and future directions. *ACM computing surveys*, *56*(5), 1-40. https://doi.org/10.1145/3625094
4. Kaur, K., Kaur, A., Gulzar, Y., & Gandhi, V. (2024). Unveiling the core of IoT: comprehensive review on data security challenges and mitigation strategies. *Frontiers in Computer Science*, *6*, 1420680. https://doi.org/10.3389/fcomp.2024.1420680
5. National Institute of Standards and Technology. (2023). Advanced Encryption Standard (AES) (FIPS 197, updated May 9, 2023). https://doi.org/10.6028/NIST.FIPS.197-upd1
6. Turan, M. S., McKay, K., Chang, D., Kang, J., & Kelsey, J. (2024). Ascon-based lightweight cryptography standards for constrained devices. In *NIST Special Publication (SP) NIST SP 800–232 ipd*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-232