Published *online* on the journal's web page : http://innovatics.unsil.ac.id

**Innovation in Research of Informatics (INNOVATICS)**

| ISSN (Online) 2656-8993 |

# A Lightweight Layered Framework for End-To-End Internet of Things Data Integrity Using Blockchain in Resource-Constrained Environments

*Eri Haryanto[1], Ahmad Nurhidayah[2]*

[1,2]*Department of Informatics, Faculty of Engineering, Universitas Janabadra, Yogyakarta 55231, Indonesia*

[1]*eri@janabadra.ac.id,* [2]*ahmadnurhidayah871@gmail.com*

ARTICLE INFORMATION

*Article History:*
Received: December 27, 2025
Last Revision: February 12, 2026
Published *Online*: March 30, 2026

KEYWORDS

Internet of Things,
Data Integrity,
Blockchain,
End-to-End Encryption,
Cybersecurity

CORRESPONDENCE

Phone: 081390252513
E-mail: *eri@janabadra.ac.id*

**ABSTRACT**

The rapid growth of Internet of Things applications has intensified the need for trustworthy data management, particularly in resource-constrained environments where sensor data are used for critical decision-making. Ensuring data integrity in such environments remains a major challenge due to the limited computational of IoT devices. This research proposes a lightweight layered framework for end-to-end IoT data integrity verification that integrates device-level encryption, centralized hashing, and blockchain-based auditing. The framework employs ESP32-based IoT devices to perform data acquisition and AES-128 end-to-end encryption, a Node.js backend server for decryption, validation, database storage, and SHA-256 hashing, and the Polygon blockchain as an immutable ledger for recording data hashes. Sensor data are encrypted directly at the device layer and transmitted securely to the backend, where integrity hashes are generated and recorded on the blockchain. Experimental evaluation was conducted by transmitting 1,000 encrypted sensor payloads from ESP32 devices to the backend system. The results show that all sensor data were successfully transmitted and processed, achieving a 100% transmission success rate, with an average backend response time of 43ms and memory utilization below 40% on IoT devices. Data manipulation experiments confirmed that all modified records were successfully detected through hash mismatch verification. However, the current evaluation is limited to a small-scale experimental setup and does not yet include large-scale multi-device deployments. Despite this limitation, the proposed framework demonstrates that strong end-to-end data integrity can be achieved without imposing significant computational overhead on resource-constrained IoT devices, making it suitable for practical IoT deployments.

## 1. INTRODUCTION

The Internet of Things (IoT) has emerged as a fundamental paradigm for large-scale and real-time data acquisition across a wide range of application domains, including smart home systems, smart agriculture, environmental monitoring, healthcare systems, and smart cities [1], [2]. In these scenarios, IoT devices are commonly deployed in distributed and unattended environments, where they operate under strict constraints in terms of processing capability, memory availability, and energy consumption. These limitations significantly complicate the deployment of comprehensive security mechanisms, making IoT systems particularly vulnerable to data manipulation, interception, and integrity violations. Consequently, ensuring data integrity and confidentiality throughout the entire IoT data lifecycle has become a critical and ongoing challenge [3], [4].

Blockchain technology has been widely recognized as a promising solution for enhancing data integrity due to its immutable, decentralized, and tamper-evident characteristics [5], [6]. By recording cryptographic hashes of data on a blockchain ledger, any unauthorized modification of the original data can be reliably detected through hash comparison. This property makes blockchain

particularly attractive as an integrity audit mechanism for data-centric systems such as IoT. However, directly integrating blockchain operations and complex cryptographic processes into IoT devices is often impractical. Blockchain interactions typically require substantial computational resources, memory, and stable network connectivity, which exceed the capabilities of most resource-constrained IoT devices [7].

To overcome these limitations, this research proposes a lightweight IoT data integrity verification framework based on a layered security architecture. In the proposed approach, data confidentiality is enforced at the device layer through lightweight end-to-end encryption, ensuring that sensor data are protected immediately at the point of generation [8]. Computationally intensive operations, including data validation, cryptographic hashing, and blockchain interaction, are delegated to a backend server with sufficient processing resources. This explicit separation of security responsibilities enables strong integrity and confidentiality guarantees while preserving the operational efficiency and longevity of IoT devices.

Despite extensive research on IoT–blockchain integration, many existing approaches tightly couple IoT devices with hashing mechanisms or blockchain clients, resulting in increased computational overhead, higher energy consumption, and reduced device lifetime. Other approaches rely on gateway-based architectures but often neglect true end-to-end data confidentiality from the data source. This research addresses these gaps by introducing a layered framework that enforces encryption at the IoT device layer, centralized hashing at the backend, and integrity on a public blockchain-based audit layer.

Building upon these limitations, an important question remains regarding how strong data integrity guarantees can be achieved in IoT systems without imposing excessive computational overhead on resource-constrained devices. In this context, this research seeks to address several key research questions: how end-to-end data integrity can be ensured in IoT environments while maintaining lightweight computational requirements for constrained devices; how encryption, hashing, and blockchain-based integrity verification can be effectively separated across different system layers to improve efficiency and scalability; and whether a layered architecture can provide reliable tamper-evident verification without requiring direct blockchain interaction on IoT devices.

The novelty of this work lies in the explicit decoupling of encryption, hashing, and blockchain anchoring across system layers. Unlike conventional IoT blockchain solutions that burden IoT devices with integrity verification or blockchain operations, the proposed framework demonstrates that strong end-to-end data integrity and auditability can be achieved without imposing significant computational overhead on resource-constrained IoT devices. The main contributions of this research are threefold: (1) the design of a lightweight layered IoT security architecture that separates encryption, hashing, and blockchain responsibilities; (2) an experimental evaluation demonstrating the feasibility of end-to-end encryption on ESP32-based IoT devices; and (3) a practical blockchain-based integrity verification mechanism suitable for real-world IoT deployments in resource-constrained environments.

## 2. RELATED WORK

The integration of Internet of Things (IoT) and blockchain technologies has been extensively explored to enhance data security, integrity, and trust in distributed environments. Christidis and Devetsikiotis [9] demonstrated how blockchain-based smart contracts can improve trust and automation in IoT ecosystems by enabling tamper-resistant data recording and decentralized control mechanisms. Similarly, Reyna et al. [10] provided a comprehensive analysis of the opportunities and challenges associated with blockchain IoT integration, emphasizing the role of immutable ledgers in ensuring data integrity and transparency. Several existing studies have focused on embedding blockchain clients or cryptographic hashing mechanisms directly into IoT devices in order to achieve decentralized integrity verification [11]. While this approach can strengthen security guarantees, it often introduces substantial computational and energy overhead, making it unsuitable for resource-constrained IoT devices with limited processing power and memory capacity. Such tight coupling between IoT devices and blockchain operations can significantly reduce device lifetime and system scalability.

Alternative approaches adopt gateway-based or cloud-centric architectures, where blockchain interactions are offloaded to intermediate nodes or backend servers. Although this strategy alleviates the computational burden on IoT devices, many implementations primarily emphasize data integrity at the storage or transmission level while overlooking true end-to-end data confidentiality from the point of data generation [12]. As a result, sensor data may still be exposed in plaintext during transmission, leaving the system vulnerable to interception and manipulation attacks. Recent research has increasingly emphasized the combination of lightweight cryptographic techniques with blockchain-based audit trails as a means of balancing security and efficiency in IoT systems [7]. Lightweight encryption algorithms have been shown to be feasible on constrained devices, while blockchain-based hash anchoring provides a reliable mechanism for integrity verification. However, practical implementations that explicitly and systematically separate encryption, hashing, and blockchain responsibilities across system layers while preserving end-to-end integrity and auditability remain limited in the literature.

This research advances the state of the art by presenting a practical and deployable IoT data integrity verification framework that enforces data confidentiality at the device layer through lightweight end-to-end encryption, centralizes hashing and validation at the backend, and anchors integrity proofs on a public blockchain-based audit layer. By decoupling security responsibilities and minimizing computational demands on IoT devices, the proposed framework addresses key limitations of existing IoT blockchain approaches and provides a scalable solution for real-world, resource-constrained IoT deployments. Several IoT system architectures, particularly in smart home implementations, adopt a backend-centric design in which sensor data are transmitted to a centralized server for processing, storage, and monitoring purposes. While this approach enables effective remote control and real-time monitoring, data

integrity verification is typically limited to database level validation without providing tamper evident or independently verifiable audit mechanisms [13].

To better illustrate the research gap identified in the literature, a comparative analysis of existing IoT security approaches is presented in Table 1.

TABLE 1. COMPARISON OF EXISTING IOT SECURITY APPROACHES

| Study | Encryption | Blockchain | End-to-End Integrity | Lightweight for IoT |
|---|---|---|---|---|
| Christidis & Devetsikiotis [9] | - | ✓ | Partial | - |
| Reyna et al. [10] | - | ✓ | Partial | - |
| Direct blockchain IoT client approaches [11] | - | ✓ | ✓ | - |
| Gateway/cloud-based IoT security frameworks [12] | Partial | Partial | Partial | ✓ |
| Lightweight cryptography for IoT [7] | ✓ | - | Partial | ✓ |
| Backend-centric IoT architectures [13] | - | - | Low | ✓ |
| Proposed Framework | ✓ | ✓ | ✓ | ✓ |

The comparison indicates that most existing studies either focus on blockchain-based integrity verification or lightweight encryption for IoT communications. However, relatively few approaches explicitly separate encryption, hashing, and blockchain anchoring across different system layers while preserving end-to-end data integrity. The proposed framework addresses this gap by introducing a layered architecture that balances security guarantees with computational efficiency for resource-constrained IoT environments.

## 3. METHODOLOGY

This research adopts an experimental and system development methodology to design, implement, and evaluate a framework for end-to-end IoT data integrity verification. The methodology is structured into several interrelated stages to clearly describe the system architecture, security mechanisms, data processing workflow, and evaluation procedures. This approach ensures that the proposed framework is reproducible, modular, and suitable for deployment in resource-constrained IoT environments. The overall research methodology is summarized in Figure 1.

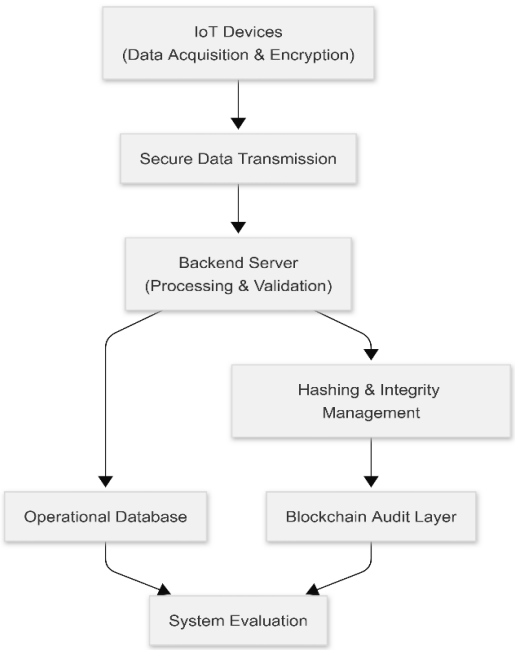


FIGURE 1. OVERVIEW OF THE PROPOSED METHODOLOGY FOR END-TO-END IOT DATA INTEGRITY VERIFICATION

### 3.1 System Architecture Design

The proposed framework is designed using a lightweight layered architecture consisting of four main components: IoT devices, a backend server, a database, and a blockchain network. Each component is assigned a distinct responsibility to enforce separation of concerns and improve system scalability and maintainability. IoT devices are responsible for sensor data acquisition and device-level encryption, ensuring data confidentiality from the point of generation. The backend server acts as a trusted processing layer that performs decryption, validation, hashing, and coordination with the blockchain network. The database serves as operational storage for sensor data and integrity-related metadata, while the blockchain functions as an immutable audit layer that records cryptographic proofs of data integrity.

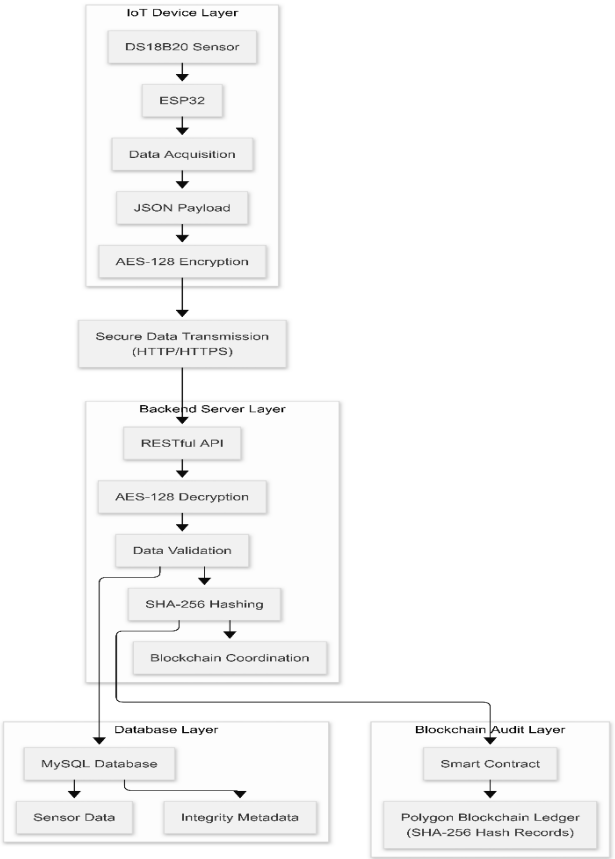


FIGURE 2. LAYERED SYSTEM ARCHITECTURE OF THE PROPOSED IOT DATA INTEGRITY VERIFICATION FRAMEWORK

This architectural design explicitly decouples encryption, hashing, and blockchain operations across different layers, thereby minimizing computational overhead on IoT devices while preserving strong end-to-end integrity guarantees. Figure 2 presents the layered system architecture of the proposed IoT data integrity verification framework, illustrating the separation of responsibilities across the IoT device, backend server, database, and blockchain audit layers. To clarify the role of each system component, Table 2 summarizes the responsibilities across architectural layers.

TABLE 2. SYSTEM COMPONENTS AND RESPONSIBILITIES

| Component | Technology | Primary Responsibility |
|---|---|---|
| IoT Device | ESP32+ DS18B20 | Data acquisition and AES-128 encryption |
| Backend Server | Node.js | Decryption, validation, hashing, blockchain coordination |
| Database | MySQL | Storage of sensor data and integrity metadata |
| Blockchain | Polygon | Immutable storage of SHA-256 hash records |

## 3.2 Security and Threat Model

To strengthen the academic rigor of the proposed framework, a threat model is defined to clarify the assumed adversarial capabilities and the security guarantees provided by the system. In this research, the adversary is assumed to have network-level access and may attempt to intercept, modify, or replay IoT data during transmission between IoT devices and the backend server. In addition, attackers may attempt to manipulate stored data within the backend database after data ingestion. However, it is assumed that the adversary cannot compromise the cryptographic keys stored within the IoT device firmware or the backend server environment.

Several potential attack scenarios are considered in this research, including data interception attacks where adversaries attempt to capture transmitted IoT payloads to access sensitive sensor information; data manipulation attacks in which attackers modify stored sensor data within the backend database after transmission; replay attacks that involve resending previously captured encrypted payloads to generate misleading or duplicated sensor records; and unauthorized data injection attacks, where malicious entities introduce fabricated sensor data into the backend system to compromise data integrity and system reliability. The proposed layered architecture mitigates these threats through multiple security mechanisms. AES-128 encryption at the IoT device layer ensures that intercepted data cannot be interpreted by attackers. SHA-256 hashing performed at the backend provides deterministic detection of any data modification. Finally, anchoring hash values on the blockchain ensures independently verifiable integrity records that cannot be altered retroactively.

Through this layered security approach, the framework provides strong guarantees for end-to-end data integrity and confidentiality while maintaining lightweight computational requirements suitable for resource-constrained IoT environments.

## *3.3 IoT Device Layer and Data Acquisition*

ESP32 microcontrollers equipped with DS18B20 temperature sensors are used as IoT devices in this research. The devices periodically acquire temperature data at fixed intervals to emulate real-time monitoring scenarios commonly found in IoT applications. The ESP32 platform is selected due to its integrated Wi-Fi capability, sufficient processing resources, and compatibility with lightweight cryptographic libraries suitable for embedded systems.

Each sensor reading is encapsulated into a structured JSON payload containing essential attributes, including a unique device identifier, the measured temperature value, and a timestamp. Structuring the data prior to encryption ensures consistency in data representation and facilitates reliable downstream processing at the backend server. The IoT payload structure is listed in Table 3.

TABLE 3. IOT PAYLOAD STRUCTURE

| Field | Data Type | Description |
|---|---|---|
| device_id | String | Unique identifier of IoT device |
| temperature | Float | Sensor measurement value |
| timestamp | Datetime | Data acquisition time |

## *3.4 End-to-End Encryption Mechanism*

To ensure data confidentiality during transmission, AES-128 symmetric encryption is implemented directly at the IoT device layer [8]. Encrypting data at the source prevents plaintext sensor values from being exposed over the network, even if communication channels are compromised. The encryption process involves payload serialization into a byte stream, application of appropriate padding, block-level encryption using AES-128, and Base64 encoding to ensure compatibility with HTTP/HTTPS transmission. By applying encryption before transmission, the framework enforces true end-to-end data confidentiality, independent of the underlying transport security mechanisms.

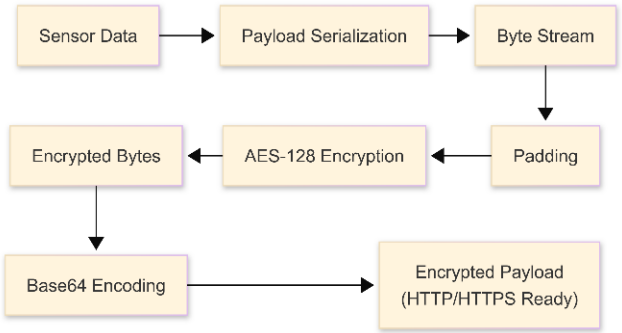


FIGURE 3. END-TO-END ENCRYPTION PROCESS AT THE IOT DEVICE LAYER USING AES-128

Figure 3 illustrates the end-to-end encryption workflow implemented at the IoT device layer, showing how sensor data are serialized, encrypted using AES-128, and encoded before being transmitted to the backend server.

## *3.5 Backend Processing and Data Validation*

Encrypted payloads transmitted by IoT devices are received by the backend server through a RESTful API interface. Upon reception, the backend performs a sequence of processing steps, including Base64 decoding, AES-128 decryption, JSON parsing, and data validation. Validation procedures ensure that the decrypted payload conforms to the expected data schema, contains valid data types, and falls within logical value ranges defined by the sensor characteristics. Payloads that fail decryption or

validation are rejected to prevent corrupted or malicious data from entering the system. Successfully validated data are then stored in a MySQL relational database for operational use and subsequent integrity verification. The sequence of backend processing steps, including decryption, parsing, validation, and database storage, is illustrated in Figure 4.

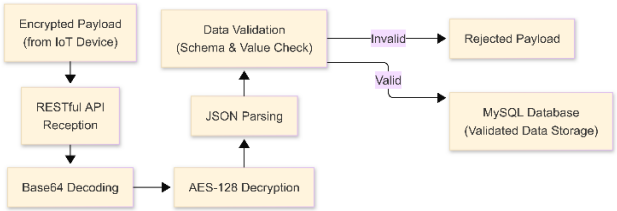


FIGURE 4. BACKEND DATA PROCESSING AND VALIDATION WORKFLOW.

### 3.6 *Integrity Verification Algorithm*

To clarify the operational workflow of the proposed framework, the integrity verification procedure implemented at the backend server is summarized in Algorithm 1. The algorithm describes the sequence of operations performed after an encrypted payload is received from an IoT device, including decoding, decryption, validation, hashing, blockchain anchoring, and integrity verification.

**Algorithm 1. IoT Data Integrity Verification Procedure**

Input: encrypted_payload
Output: integrity_status

1. Receive encrypted payload from IoT device
2. Decode Base64 payload
3. Decrypt payload using AES-128
4. Parse JSON sensor data
5. Validate data structure and value range
6. Store validated data in database
7. Generate SHA-256 hash from stored data
8. Submit hash value to blockchain smart contract
9. Retrieve stored hash from blockchain
10. Compare computed hash with blockchain hash

If hash_match = true
  integrity_status = VALID
Else
  integrity_status = TAMPERED
End If

### 3.7 *Hashing and Integrity Management*

After successful validation and storage, the backend server generates a cryptographic hash of the sensor data using the SHA-256 algorithm [14]. Hashing is centralized at the backend layer to ensure consistency in hash generation and to reduce computational and energy overhead on IoT devices. The hash value serves as a compact and tamper-sensitive representation of the original data. Any modification to the stored data, regardless of scale, produces a different hash value, enabling deterministic detection of data manipulation. Centralized hashing also allows the integration of additional metadata into the hashing process, further strengthening integrity guarantees.

### 3.8 *Blockchain Integration and Audit Layer*

The generated SHA-256 hash values are recorded on the Polygon blockchain through a smart contract interface. To preserve data privacy and minimize blockchain storage and transaction costs, only hash values and minimal metadata are stored on the blockchain, while raw sensor data remain in the backend database.

Blockchain transactions are executed asynchronously to prevent increased latency in the primary data ingestion pipeline. This design choice ensures that temporary delays or failures in blockchain communication do not disrupt real-time IoT data acquisition and processing. The blockchain thus functions as an immutable and independently verifiable audit layer that enables long-term integrity verification [15].

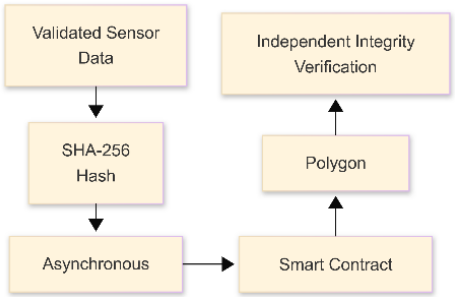


FIGURE 5. BLOCKCHAIN-BASED AUDIT LAYER FOR IOT DATA INTEGRITY VERIFICATION

Figure 5 illustrates the blockchain-based audit layer, showing how SHA-256 hash values are anchored on the Polygon blockchain while raw sensor data are retained off-chain, ensuring data integrity without on-chain data storage overhead.

### 3.9 *System Evaluation Procedure*

System evaluation is conducted through functional testing, end-to-end integrity verification scenarios, and data manipulation experiments. Functional testing assesses the reliability of data transmission, decryption, validation, and storage processes. End-to-end integrity verification evaluates the consistency between backend-generated hashes and blockchain records. Data manipulation tests confirm the system's ability to detect unauthorized data modifications through hash mismatch analysis.

TABLE 4. EVALUATION METRICS

| Metric | Description |
|---|---|
| Transmission success rate | Percentage of payloads successfully processed |
| Backend response time | Processing latency per payload |
| Encryption overhead | Impact of AES-128 on IoT performance |
| IoT memory usage | RAM utilization on ESP32 |
| Integrity detection accuracy | Ability to detect data manipulation |

Table 4 summarizes the evaluation metrics applied in functional testing, integrity verification, and data manipulation experiments.

## 4. RESULT AND DISCUSSION

The proposed IoT data integrity verification framework was experimentally evaluated to assess its reliability, performance, and effectiveness under continuous data transmission scenarios. The evaluation focused on functional correctness, end-to-end integrity assurance, system performance, and resource utilization on constrained IoT devices.

### 4.1 *Functional Testing and System Reliability*

Functional testing was conducted by transmitting encrypted sensor data from ESP32-based IoT devices to the backend server over a continuous period. A total of

1,000 encrypted payloads were generated and sent using a fixed transmission interval. Experimental results indicate that all payloads were successfully received, decrypted, validated, and stored by the backend server, resulting in a transmission success rate of 100%.

The average backend response time, measured from payload reception to successful database storage, was 43 ms. This latency includes Base64 decoding, AES-128 decryption, JSON parsing, validation, and database insertion. The observed response time demonstrates that the backend processing pipeline is efficient and suitable for real-time IoT data ingestion. Table 5 summarizes the results of functional testing and overall system reliability.

TABLE 5. FUNCTIONAL TESTING AND SYSTEM PERFORMANCE SUMMARY

| Metric | Result |
|---|---|
| Total payloads transmitted | 1,000 |
| Successful transmissions | 1,000 (100%) |
| Average backend response time | 43 ms |
| Payload loss | 0 |
| Processing errors | 0 |

These results confirm that the proposed framework operates reliably under continuous data transmission conditions and does not introduce instability or bottlenecks in the primary data ingestion process.

### 4.2 *End-to-End Integrity Verification Results*

End-to-end integrity verification was performed by comparing SHA-256 hash values generated at the backend with the corresponding hash records stored on the blockchain. For all unmodified sensor data, the computed hash values matched exactly with the hashes anchored on the blockchain, confirming the correctness of the integrity verification mechanism. To further evaluate system robustness, data manipulation tests were conducted by intentionally altering selected records in the backend database after hash anchoring. In all manipulation scenarios, the recalculated hash values differed from the blockchain-stored hashes, allowing the system to detect integrity violations deterministically and without ambiguity.

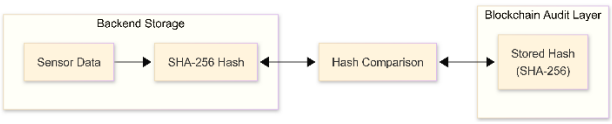


FIGURE 6. HASH-BASED INTEGRITY VERIFICATION BETWEEN BACKEND STORAGE AND BLOCKCHAIN AUDIT LAYER

The integrity verification workflow, including the comparison between backend-generated SHA-256 hashes and immutable blockchain records, is illustrated in Figure 6. This mechanism enables reliable detection of unauthorized data modifications without requiring raw sensor data to be stored on the blockchain. The results of integrity verification and data manipulation detection are summarized in Table 6.

TABLE 6. INTEGRITY VERIFICATION AND MANIPULATION DETECTION

| Test Scenario | Number of Samples | Detection Accuracy |
|---|---|---|
| Unmodified data | 50 | 100% valid |
| Modified data | 10 | 100% detected |
| False positives | – | 0 |
| False negatives | – | 0 |

The absence of false positives and false negatives demonstrates that the hash-based verification mechanism provides reliable and deterministic integrity assurance.

### 4.3 *Resource Utilization and Encryption Overhead*

The impact of device-level encryption on IoT resource utilization was evaluated by monitoring memory usage and processing behavior on ESP32 devices during continuous operation. The AES-128 encryption process consumed less than 40% of available RAM and did not introduce noticeable delays in sensor data acquisition or transmission cycles. Figure 7 shows the resource utilization on ESP32 devices during AES-128 encryption and data transmission, demonstrating that the encryption process introduces minimal overhead on constrained hardware.

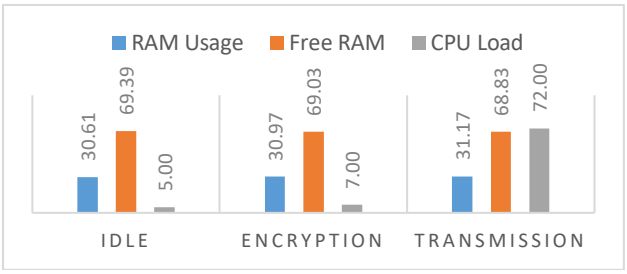


FIGURE 7. RESOURCE UTILIZATION ON ESP32 DEVICES DURING AES-128 ENCRYPTION AND DATA TRANSMISSION.

The results demonstrate that the proposed layered framework effectively limits computational overhead on the IoT device, with encryption introducing only minimal CPU and memory usage while the primary performance cost arises from secure data transmission. These findings align with previous studies on lightweight cryptography for resource-constrained IoT environments [7], confirming that AES-128 is feasible for device-level end-to-end encryption without degrading overall system performance. By isolating computationally intensive blockchain operations from the IoT device and executing them at the backend, the framework preserves device efficiency while maintaining strong security guarantees.

### 4.4 *Comparative Security Evaluation*

To highlight the advantages of the proposed framework, a comparative evaluation was conducted by analyzing different security configurations commonly used in IoT systems.

TABLE 7. COMPARATIVE SECURITY EVALUATION

| Approach | Encryption | Hashing | Blockchain | Integrity Protection |
|---|---|---|---|---|
| No Security Mechanism | – | – | – | Very Low |
| Encryption Only | ✓ | – | – | Medium |
| Hashing Only | – | ✓ | – | Medium |
| Encryption + Hashing | ✓ | ✓ | – | High |
| Proposed Framework | ✓ | ✓ | ✓ | Very High |

The comparison shows that systems relying only on encryption or hashing provide partial protection against attacks. Encryption ensures confidentiality but does not

guarantee integrity verification, while hashing alone cannot protect data during transmission. By combining device-level encryption, backend hashing, and blockchain anchoring, the proposed framework provides stronger end-to-end integrity guarantees and tamper-evident verification compared with conventional approaches.

### *4.5 Discussion*

The experimental results demonstrate that the proposed framework effectively achieves end-to-end data integrity verification while remaining efficient and scalable for resource-constrained IoT deployments. Centralizing hashing and blockchain interactions at the backend significantly reduces computational and energy overhead on IoT devices, addressing key limitations identified in existing IoT–blockchain approaches. Moreover, the asynchronous anchoring of hash values to the blockchain ensures that blockchain latency does not interfere with real-time data ingestion, making the framework suitable for continuous monitoring applications. The combination of device-level encryption, centralized integrity management, and blockchain-based auditability provides a balanced and practical solution for trustworthy IoT data management.

From a practical deployment perspective, the operational characteristics of the blockchain network, including transaction cost and confirmation latency, must also be considered. In this research, the Polygon blockchain network was selected due to its relatively low transaction fees and higher throughput compared with many other public blockchain platforms. In general, recording integrity hashes on the blockchain requires only a small transaction fee, making the approach economically feasible for many IoT monitoring scenarios. Furthermore, blockchain confirmation latency may vary depending on network conditions. However, because hash anchoring in the proposed framework is performed asynchronously at the backend layer, blockchain confirmation time does not interfere with the real-time ingestion and processing of IoT data. This design allows the blockchain to function as an independent audit layer that strengthens data integrity assurance without introducing performance bottlenecks in the primary IoT data processing pipeline.

## 5. CONCLUSIONS

This research proposes a lightweight layered framework for end-to-end IoT data integrity verification in resource-constrained environments. The architecture separates encryption, hashing, and blockchain anchoring across different system layers, enabling strong security guarantees without imposing excessive computational overhead on IoT devices. Experimental results indicate reliable system operation, achieving a 100% transmission success rate, an average backend processing latency of approximately 43 ms, and memory usage below 40% of available RAM on ESP32 devices. Data manipulation experiments further confirm that unauthorized modifications can be deterministically detected through SHA-256 hash mismatch verification against blockchain records. The main contribution of this research is the design and experimental validation of a layered security architecture that decouples encryption, hashing, and blockchain operations, enabling trustworthy IoT data integrity verification without requiring direct blockchain interaction on resource-constrained devices. Future work will evaluate the framework in large-scale multi-device deployments, integrate heterogeneous sensor systems, and explore smart contract–based automation for real-time integrity monitoring.

## REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm," *Ad Hoc Networks*, vol. 56, pp. 122–140, Mar. 2019, doi: 10.1016/j.adhoc.2016.12.004.

[2] Ruuhwan, Randi Rizal, and Indra Karyana, "Sistem Kendali dan Monitoring pada Smart Home Berbasis Internet of Things (IoT)," vol. 1, no. 2, pp. 43–50, 2019.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy & Trust in Internet of Things: the road ahead."

[4] E. Haryanto and I. Riadi, "Forensik Internet of Things Pada Device Level Berbasis Embedded System," vol. 6, no. 6, 2019, doi: 10.25126/jtiik.201961465.

[5] Desi Mutiara Azizah and Caca Oktavia, "Implementing Blockchain Technology for Securing IoT-Based Smart Grids," *International Journal of Electrical Engineering, Mathematics and Computer Science*, vol. 1, no. 1, pp. 09–12, Mar. 2024, doi: 10.62951/ijeemcs.v1i1.70.

[6] A. K. Vedantham, "Blockchain Technology: Advancing Data Integrity and Security in Modern Systems," *International Journal of Research In Computer Applications and Information Technology (IJRCAIT)*, vol. 7, no. 2, pp. 1574–1582, doi: 10.5281/zenodo.14232867.

[7] Amrita, C. P. Ekwueme, I. H. Adam, and A. Dwivedi, "Lightweight Cryptography for Internet of Things: A Review," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024, doi: 10.4108/eetiot.5565.

[8] R. Kurniawan Endrayanto, A. Muttaqin, and R. A. Setyawan, "Advanced Encryption Standard (AES) pada Modul Internet of Things (IoT) Advanced Encryption Standard (AES) on Internet of Things (IoT) Module," *TELKA*, vol. 5, no. 2, pp. 103–113, 2019.

[9] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," 2016, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2016.2566339.

[10] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018, doi: 10.1016/j.future.2018.05.046.

[11] S. Albaqami, M. Nekovee, and I. Khan, "Blockchain in IoT: Applications, Challenges, and

Future Directions," in *Lecture Notes in Networks and Systems*, Springer, 2024.

[12] S. Alharbi, W. Awad, and D. Bell, "HECS4MQTT: A Multi-Layer Security Framework for Lightweight and Robust Encryption in Healthcare IoT Communications," *Future Internet*, vol. 17, no. 7, Jul. 2025, doi: 10.3390/fi17070298.

[13] R. Rizal, S. R. Selamat, M. Z. Mas'ud, and N. Widiyasono, "Enhanced Readiness Forensic Framework for the Complexity of Internet of Things (IoT) Investigation Based on Artificial Intelligence," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 50, no. 1, pp. 121–135, Aug. 2025, doi: 10.37934/araset.50.1.121135.

[14] William. Stallings, *Cryptography and network security : principles and practice*. Pearson, 2020.

[15] M. Bjelic, S. Nailwal, A. Chaudhary, and W. Deng, *POL: One token for all Polygon chains.* Polygon Labs, White Paper, 2019.

## AUTHORS

**Eri Haryanto**
Lecturer in the Department of Informatics, Faculty of Engineering, Janabadra University, Indonesia. His research interests focus on Internet of Things (IoT), IoT security, digital forensics, and web programming, particularly in the development of secure and lightweight IoT systems.

**Ahmad Nurhidayah**
Student in the Department of Informatics, Faculty of Engineering, Janabadra University, Indonesia.