International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

Securing IIoT Environments with Blockchain-Enabled End-to-End Protection

Qudsia Zainab[1], Dr. Md Zainlabuddin[2]
[1]PG Scholar, Department of CSE, ISL Engineering College, Hyderabad, India.
[2]Associate Professor, Department of CSE, ISL Engineering College, Hyderabad, India.
Email ID: qudsiazainab003@gmail.com

ABSTRACT
The rapid growth of the Industrial Internet of Things (IIoT) has significantly enhanced industrial efficiency, productivity, automation, and operational performance. However, conventional security frameworks based on static policies and centralized management are increasingly inadequate for the scale, heterogeneity, and dynamic nature of modern IIoT environments. These approaches face challenges related to scalability, data integrity, authentication, and adaptability, thereby increasing the vulnerability of industrial systems to cyberattacks. To address these limitations, this research proposes an innovative end-to-end security framework based on blockchain technology for IIoT environments. The proposed framework employs a decentralized architecture that eliminates dependence on a single central authority while providing secure authentication, authorization, and data integrity across distributed industrial networks. Smart contracts are incorporated to enable automated and dynamic enforcement of security policies, allowing rapid adaptation to emerging security threats. In addition, lightweight cryptographic mechanisms are utilized to reduce computational and communication overhead, making the framework suitable for resource-constrained IIoT devices. A hybrid blockchain architecture combining public and private blockchain characteristics is introduced to balance security, scalability, privacy, and performance. The framework further improves data efficiency by storing only essential information on the blockchain while maintaining reliable and verifiable records. Overall, the proposed approach provides a resilient, scalable, adaptive, and secure solution for protecting heterogeneous IIoT ecosystems against evolving cyber threats.

Keywords: *Industrial Internet of Things (IIoT), Blockchain Technology, Cybersecurity, Decentralized Security, Smart Contracts, Authentication, Authorization, Data Integrity, Lightweight Cryptography, Hybrid Blockchain, Scalability, Data Security.*

1. Introduction

The Industrial Internet of Things (IIoT) has become an important technological foundation for modern industrial environments by connecting machines, sensors, controllers, production systems, and analytical platforms through networked communication infrastructures. The continuous exchange of operational information enables organizations to improve production planning, equipment utilization, process monitoring, predictive maintenance, and decision-making. The integration of IIoT with emerging technologies such as artificial intelligence, edge computing, cloud computing, digital twins, and Industry 4.0 platforms has further expanded the capabilities of connected industrial systems [2], [3], [6], [7]. At the same time, the increasing dependence on interconnected devices has expanded the industrial attack surface and introduced security challenges that are considerably more complex than those encountered in isolated automation environments.

Traditional industrial control and monitoring architectures generally depend on centralized servers, predefined access policies, and trusted administrative entities. Although these mechanisms have supported industrial operations for many years,

Received: *11-06-2026*
Revised: *22-07-2026*
Accepted: *10-08-2026*
Published: *18-08-2026*
Citation: *"Securing IIoT Environments with Blockchain-Enabled End-to-End Protection", ijaicn, vol. 2, no. 3, pp. 45–60, August. 2026,*

Their effectiveness becomes increasingly limited when thousands of heterogeneous IIoT devices communicate across distributed environments. Centralized security mechanisms can create attractive targets for attackers because compromising a central authority or management server may provide access to a large portion of the connected infrastructure.

Data generated by IIoT devices represents another significant security concern. Industrial sensors continuously produce information relating to machine conditions, production processes, energy consumption, environmental parameters, and operational status. Unauthorized modification of such information can result in incorrect decisions, production interruptions, equipment damage, or safety-related consequences. Therefore, maintaining data integrity and establishing confidence in the origin of industrial information are fundamental requirements. Existing research has investigated reliable device-access mechanisms and security frameworks for IIoT environments, demonstrating the importance of dependable authentication and controlled access to distributed industrial resources [16].

The rapid expansion of IIoT networks has also increased the requirement for effective intrusion detection and attack identification mechanisms. Industrial networks may contain devices with different computational capabilities, communication protocols, operating characteristics, and security requirements. Consequently, a single static security policy may not be sufficiently effective for all devices and operational conditions. Software-defined networking has been investigated as a means of improving the flexibility of industrial intrusion detection by providing programmable network control and centralized visibility over distributed communication flows [4]. However, approaches that continue to rely heavily on centralized control may still face scalability, availability, and trust-related limitations.

Blockchain technology offers an alternative approach by introducing distributed trust, tamper-evident record keeping, cryptographic verification, and decentralized transaction management. Instead of relying entirely on a single trusted authority, blockchain enables participating entities to maintain a shared and verifiable representation of selected security and operational events. Its application to IIoT has received considerable attention, particularly for establishing trust between industrial devices and protecting interactions within distributed environments. Blockchain-based trust mechanisms have been investigated in digital-twin-enabled IIoT systems, demonstrating how distributed ledgers can contribute to trustworthy interactions among interconnected industrial entities [9].

The integration of blockchain with artificial intelligence can provide additional security capabilities. Deep learning techniques can assist in identifying complex or previously unseen attack patterns, while blockchain can provide a verifiable mechanism for recording relevant security events and maintaining trusted information. A deep-learning-integrated blockchain framework for industrial IoT illustrates the potential of combining intelligent threat analysis with distributed security mechanisms [14]. Such integration can support more comprehensive protection by addressing both the detection and trust dimensions of IIoT security.

Nevertheless, directly applying conventional blockchain architectures to industrial environments introduces several challenges. IIoT devices frequently operate under strict limitations in processing capability, memory, energy consumption, and communication bandwidth. Maintaining a complete distributed ledger on every device may therefore be impractical. Furthermore, consensus procedures and excessive transaction processing can introduce latency and communication overhead. Edge–fog–cloud architectures have been studied as a means of distributing IIoT data processing across different computational layers, thereby improving scalability and reducing the burden placed on individual devices [10]. These architectural principles are relevant to blockchain-enabled IIoT security because security operations can similarly be distributed according to device capabilities and network requirements.

Another important issue is the balance between security, privacy, and performance. Industrial organizations may need to protect confidential production information while simultaneously maintaining auditable records of security-related events. A completely public blockchain may not be suitable for sensitive industrial information, whereas a completely private architecture can reduce some of the decentralization benefits associated with blockchain. A hybrid architecture can therefore provide a practical compromise by restricting sensitive information to authorized participants while retaining distributed verification mechanisms for selected transactions.

The increasing adoption of federated learning and other distributed intelligence approaches further demonstrates the movement toward decentralized processing in IIoT. Federated learning enables participating devices or edge nodes to contribute to model development without necessarily transferring their complete local datasets to a centralized server. Such approaches can support privacy-conscious intelligence in distributed industrial systems [12].

Combining decentralized learning with blockchain-based trust management may therefore provide a foundation for security architectures in which device identity, data integrity, access decisions, and threat intelligence can be coordinated without excessive dependence on centralized infrastructure.

Based on these challenges, this research proposes a blockchain-enabled end-to-end protection framework for IIoT environments. The proposed approach focuses on decentralized authentication, controlled authorization, data integrity verification, smart-contract-based policy enforcement, lightweight cryptographic protection, and selective blockchain storage. A hybrid blockchain architecture is considered to accommodate the security and privacy requirements of heterogeneous industrial environments while limiting unnecessary computational and communication overhead. The overall objective is to establish a resilient security framework capable of protecting IIoT communication and data throughout their lifecycle while maintaining the scalability and operational efficiency required by modern industrial systems.

2. 2. Literature Review

2.1 Industrial IoT Architecture and Security Challenges

Peter, Pradhan, and Mbohwa examined the opportunities and requirements associated with the Industrial Internet of Things in manufacturing environments, particularly within emerging economies. Their work emphasizes that IIoT can improve industrial productivity and operational efficiency through connected devices and intelligent information exchange. At the same time, the authors identify infrastructure, interoperability, resource, and security challenges that must be addressed for successful industrial adoption. Their findings establish the broader technological context in which secure IIoT architectures must operate and demonstrate why security cannot be considered separately from scalability and industrial infrastructure requirements [2].

Babayigit and Abubaker investigated the development of IIoT systems in comparison with traditional SCADA-based industrial automation. Their study highlights the transition from relatively isolated control environments toward highly connected and distributed industrial architectures. This transformation creates new communication paths and increases the number of potentially vulnerable endpoints. The work consequently supports the need for security mechanisms that can operate across heterogeneous devices while maintaining the reliability and availability expected from industrial automation systems [6].

Singh, Ahuja, Singh, and Singh explored the application of Quality 4.0 and IIoT within agricultural manufacturing. Their research demonstrates how connected industrial technologies can contribute to improved quality management, monitoring, and decision-making. However, the growing dependence on digital connectivity also means that industrial information must be protected against unauthorized access and manipulation. The study is relevant to the present research because it illustrates how IIoT security must support not only traditional cybersecurity objectives but also the operational requirements of data-driven quality and manufacturing systems [7].

2.2 Distributed IIoT Processing and Scalability

Kumar and Agrawal presented a survey of multidimensional IIoT data processing within edge–fog–cloud architectural frameworks. Their work discusses the increasing volume and complexity of industrial data and the role of distributed computing layers in handling this information efficiently. Edge and fog processing can reduce communication delays and decrease dependence on centralized cloud resources. These concepts are particularly significant for blockchain-based security because security-related processing can be positioned closer to industrial devices instead of requiring every operation to be performed by a remote centralized service [10].

Truong, Ha, Nayyar, Bilal, and Kwak investigated performance optimization for IIoT devices using radio-frequency energy harvesting, NOMA, and mobile edge computing. Their study demonstrates the importance of energy efficiency and computational optimization in resource-constrained IIoT environments. Industrial security mechanisms must account for these constraints because complex cryptographic procedures and distributed consensus operations may impose additional resource requirements. Their findings reinforce the need for lightweight and carefully optimized security mechanisms when blockchain is introduced into IIoT networks [13].

Kumar, Walia, Shingare, Singh, and Gill proposed an AI-based sustainable offloading framework for IIoT operating across collaborative cloud–fog environments. Their work demonstrates the potential of intelligent task distribution to improve the efficiency of distributed industrial computing. From a security perspective, such architectures provide opportunities for placing authentication, monitoring, and security analytics at appropriate computational layers. This supports the development of blockchain-enabled security systems in which resource-intensive functions can be delegated to capable edge or fog nodes while constrained devices perform only essential operations [20].

2.3 Authentication, Access Control, and Intrusion Detection

Alasmary introduced RDAF-IIoT, a reliable device-access framework designed for Industrial Internet of Things environments. The study emphasizes the importance of reliable access mechanisms for controlling interactions between industrial devices and network resources. Authentication and authorization are especially important in IIoT because unauthorized devices or compromised endpoints may become entry points for attacks. The research provides a foundation for incorporating stronger identity and access-management mechanisms into distributed industrial security architectures [16].

Alshahrani, Khan, Rizwan, Reshan, Sulaiman, and Shaikh developed an intrusion detection framework for IIoT using software-defined networking. Their work demonstrates how programmable network architectures can provide greater flexibility for monitoring industrial traffic and identifying malicious behavior. The use of SDN can improve network visibility and enable security policies to be modified according to changing conditions. Nevertheless, dependence on centralized network control introduces additional trust and availability considerations, which can motivate the use of distributed mechanisms such as blockchain for maintaining verifiable security records [4].

M. D. Zainlabuddin and N. Sharma investigated security enhancement in data propagation for wireless networks. Their research addresses the protection of information during communication and highlights the significance of secure data transmission in distributed network environments. Although the study is not limited to IIoT, its focus on secure propagation is relevant to industrial networks in which sensor information and control-related messages frequently traverse wireless communication channels. The principles of secure propagation can complement blockchain-based integrity verification within an end-to-end IIoT protection architecture [24].

2.4 Blockchain-Based Security and Trust

Sasikumar, Vairavasundaram, Kotecha, Indragandhi, Ravi, Selvachandran, and Abraham investigated a blockchain-based trust mechanism for digital-twin-enabled IIoT systems. Their work demonstrates how blockchain can support trust relationships among distributed industrial entities while maintaining verifiable records of interactions. The integration of blockchain with digital twins is particularly significant because digital twins depend on reliable data to represent physical assets accurately. Their study provides strong support for using distributed ledger technology as a trust layer within complex industrial ecosystems [9].

Aljuhani, Kumar, Alanazi, Albalawi, Taouali, Islam, Kumar, and Alazab proposed a deep-learning-integrated blockchain framework for securing industrial IoT. Their research combines intelligent analysis with distributed ledger mechanisms, demonstrating that blockchain can provide trusted data management while deep learning can assist in identifying security threats. This combination is relevant to the proposed framework because effective IIoT protection requires both reliable security records and intelligent identification of abnormal or malicious activities [14].

Uddin and Zainlabuddin examined secure video processing and watermark embedding using a Shamir secret-sharing-based approach. Their work demonstrates the use of cryptographic and secret-sharing principles for protecting digital information and controlling access to sensitive data. While their research focuses on secure multimedia processing rather than industrial IoT, the underlying concepts of cryptographic protection and controlled information reconstruction are relevant to the development of lightweight security mechanisms for IIoT data and communication [1].

2.5 Intelligent and Adaptive Industrial Security

Kumar, Pawar, Gonaygunta, and Singh reviewed the impact of federated learning on industrial IoT. Their study discusses how federated learning can enable collaborative model development while reducing the requirement to centrally collect raw data. This approach is particularly useful in industrial environments where operational datasets may contain sensitive information. Federated learning can complement blockchain by using the ledger to establish trust, record model-related transactions, or verify participating entities while distributed learning provides intelligent security analysis [12].

Arputharaj and Pal examined real-time monitoring and decision-making in the context of Industry 5.0 and IIoT. Their work highlights the increasing importance of real-time industrial intelligence and human-centered decision support. As industrial systems become more autonomous and responsive, security mechanisms must also operate with limited latency. A blockchain-enabled framework therefore needs to balance the advantages of immutable verification with the response-time requirements of real-time industrial operations [17].

K. S. Hawaou, V. C. Kamla, S. Yassa, O. Romain, J. E. N. Mboula, and L. Bitjoka surveyed Industry 4.0 workflow scheduling and examined the challenges associated with coordinating industrial processes in increasingly interconnected environments. Their work illustrates the complexity of scheduling and resource allocation when multiple industrial components and computational services interact dynamically. From the perspective of

cybersecurity, security operations must be integrated into these dynamic workflows without creating unnecessary delays or resource bottlenecks. This supports the need for adaptive security policies and automated enforcement mechanisms in blockchain-enabled IIoT environments [22].

2.6 Research Gap

The reviewed literature demonstrates substantial progress in IIoT architecture, distributed computing, authentication, intrusion detection, blockchain-based trust, and intelligent security. However, many existing approaches concentrate on individual security requirements rather than providing a unified end-to-end protection mechanism. Blockchain-based solutions can improve trust and integrity, but their deployment in resource-constrained IIoT environments may introduce computational, communication, storage, and latency challenges. Similarly, AI-based security mechanisms can improve threat detection but require reliable data and trustworthy coordination among participating devices.

The major research gap therefore lies in developing an integrated security architecture that combines decentralized identity management, authentication, authorization, data integrity, adaptive security policies, lightweight cryptography, and intelligent threat management while maintaining the performance requirements of industrial environments. The proposed research addresses this gap through a hybrid blockchain-enabled framework in which only essential security information is maintained on-chain, computationally intensive operations can be delegated to suitable edge or fog resources, and smart contracts can automate security-policy enforcement. Such an approach aims to establish a practical balance between decentralization, security, privacy, scalability, and industrial performance.

Proposed System

The proposed system integrates lightweight cryptographic security mechanisms with blockchain technology to provide secure, efficient, and reliable management of IoT-based weather data. The primary purpose of the framework is to protect weather information generated by IoT sensors during collection, transmission, storage, and retrieval. Since IoT devices generally operate with limited computational power, memory, storage capacity, and energy resources, the proposed approach emphasizes lightweight security mechanisms that provide adequate protection without significantly affecting device performance. The system follows a secure processing sequence in which weather information is collected from IoT devices, preprocessed, encrypted, integrity-protected, and submitted to the blockchain network. A hybrid cryptographic approach can be used in which a symmetric encryption mechanism protects the actual weather-data payload, while an asymmetric mechanism such as Elliptic Curve Cryptography (ECC) is used for secure key establishment or key protection. This approach reduces the computational burden associated with encrypting large amounts of data using asymmetric cryptography. After encryption, a cryptographic hash is generated to provide integrity verification, and the protected information is submitted to the blockchain as a transaction. The blockchain maintains a tamper-evident record of security-relevant information and provides traceability for subsequent verification.

The proposed system also incorporates smart contracts to automate access control, data validation, and security-policy enforcement. When an authorized user or application requests access to weather information, the smart contract evaluates the identity, role, permissions, and applicable security policies before producing an access decision. The same mechanism can be used to verify data integrity and identify abnormal or inconsistent sensor observations according to predefined rules. This reduces dependence on manual intervention and provides consistent execution of security policies. The framework is designed to support scalability as the number of IoT devices increases. Instead of storing every item of raw weather information directly on the blockchain, only essential transaction information, hashes, identifiers, timestamps, and security metadata need to be maintained on-chain, while larger datasets can be maintained through suitable external storage. This hybrid storage strategy reduces blockchain growth and improves system efficiency. The performance of the proposed framework can be evaluated using encryption time, decryption time, transaction latency, throughput, computational overhead, and energy consumption. Lightweight cryptographic mechanisms, including algorithms designed for constrained environments, can contribute to lower energy requirements compared with computationally intensive conventional security techniques. The resulting architecture therefore aims to establish a balance among confidentiality, integrity, authentication, scalability, performance, and energy efficiency.

3. Methodology

The methodology of the proposed system consists of interconnected modules that collectively provide secure acquisition, protection, storage, validation, visualization, and management of IoT weather information. The major components include the IoT Weather Data Acquisition module, Blockchain

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

Layer, Smart Contract Engine, Cryptographic Security module, JSP Dashboard, and MySQL Database. The IoT Weather Data Acquisition module is responsible for collecting environmental parameters from sensors or simulated data sources. The Blockchain Layer provides distributed and tamper-evident storage of security-relevant records. The Smart Contract Engine automates data validation and access-control decisions. The Cryptographic Security module protects information using lightweight encryption and integrity mechanisms. The JSP Dashboard provides an interface for users and administrators to interact with the system, while the MySQL Database maintains supporting information such as user accounts, roles, device metadata, and application logs. The integration of these components produces a complete security workflow in which data generated at the IoT layer is protected before entering the blockchain environment and is subsequently made available only to authorized users.

Techniques and Algorithms Used

The Decentralized Hybrid Blockchain Algorithm forms the principal architectural mechanism of the proposed security framework. It is designed to provide secure collection, verification, and management of IoT weather data without relying exclusively on a centralized authority. Weather information generated by IoT sensors is first identified using a unique device identifier and timestamp. The information is then processed through the cryptographic layer, where appropriate encryption and integrity mechanisms are applied. The protected transaction is submitted to authorized blockchain nodes for validation. Once the transaction satisfies the required blockchain rules, it is incorporated into the distributed ledger. The cryptographic linking of records provides a mechanism for detecting unauthorized modification and establishes a traceable history of data transactions. The hybrid nature of the architecture combines controlled participation with decentralized verification. Since weather information may be sensitive and the participating organizations may need to be identified, the blockchain environment can be implemented as a permissioned or private network. At the same time, the distributed ledger provides the verification and tamper-evident characteristics associated with blockchain systems. Smart contracts operate within this architecture to automate data validation and access control. They can evaluate device status, user permissions, transaction attributes, and predefined policies before accepting or authorizing operations. The algorithm is therefore designed to provide security, transparency, traceability, and controlled access while maintaining an architecture suitable for IoT environments.

Requirements Engineering

Hardware Requirements

The proposed system can be implemented using a conventional computer system capable of running a Java-based web application, MySQL database, blockchain services, and cryptographic operations. A minimum configuration of an Intel Core i3 processor or equivalent, 4 GB of RAM, and at least 100 GB of available storage is sufficient for basic development and testing. A 15.6-inch LED monitor or equivalent display can be used for application development and dashboard visualization. Standard input devices such as a keyboard and mouse are required. When physical IoT devices are used, appropriate sensors, microcontrollers, communication modules, and networking equipment are additionally required. For experiments involving multiple blockchain nodes or simultaneous application services, higher processor and memory capacity is recommended to prevent resource contention.

Software Requirements

The software environment consists of Java-based web technologies, database services, blockchain components, development tools, and client-side browser support. The front-end interface can be developed using JavaServer Pages, HTML, CSS, and JavaScript, while Java Servlets can provide server-side application logic. MySQL 5.5 or a later compatible version can be used for supporting data management. The application can operate on Windows 10 or Windows 11 during development and testing, while Apache Tomcat can provide the web-application runtime environment. Google Chrome, Mozilla Firefox, or Microsoft Edge can be used to access the JSP dashboard. Eclipse IDE or IntelliJ IDEA can be used for software development. Cryptographic operations can be implemented using Java Cryptography Architecture or an equivalent security library, while the blockchain component can be implemented using an appropriate permissioned or private blockchain platform.

4. Design Engineering

Design engineering provides a structured representation of the proposed **Secure IoT Weather Data Management System Using Blockchain and Lightweight Cryptography**. It defines the architecture, functional behavior, data relationships, communication patterns, and deployment structure of the system. The design is developed to support secure acquisition, encryption, validation, blockchain-based storage, controlled access, and visualization of IoT weather information. Since IoT environments involve numerous heterogeneous devices with limited computational and energy

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

resources, the proposed design separates computationally intensive operations from device-level activities wherever possible. Lightweight cryptographic mechanisms are applied before data transmission, while blockchain provides an immutable record of verified transactions. Smart contracts are incorporated to automate validation and access-control decisions, thereby reducing dependence on manual administration. The system design is represented through several UML models, including use case, class, object, state chart, sequence, collaboration, activity, component, and deployment diagrams. Each model describes a particular perspective of the proposed solution. The use case model identifies interactions among IoT devices, users, and administrators. The class model defines the principal software entities and their relationships. Sequence and collaboration models describe the communication among system components during data processing. The activity and state models represent operational workflows and data-state transitions. The component and deployment diagrams describe the logical and physical organization of the software and hardware infrastructure. Together, these design models provide a comprehensive blueprint for implementing a secure, scalable, and maintainable IoT weather-data management platform.

System Architecture

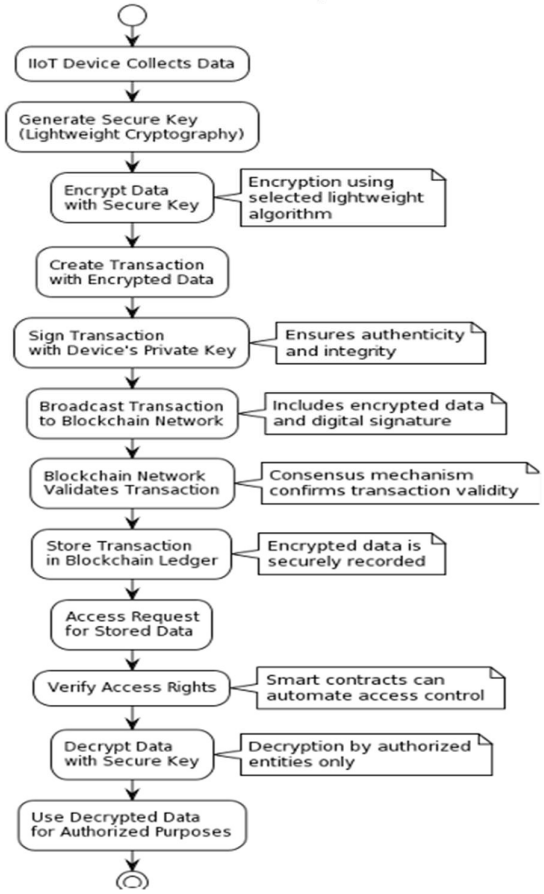

Figure1: System Architecture

The proposed system architecture combines IoT data acquisition, lightweight cryptography, blockchain technology, smart contracts, database services, and a web-based dashboard. At the first layer, IoT devices or weather-data sources collect environmental measurements. Because many IoT devices have limited processing capability, the security mechanism is designed to minimize computational, memory, and energy requirements. The collected information is therefore subjected to lightweight cryptographic processing before it is transmitted to the blockchain environment. The protected transaction is examined by the smart contract layer, which automatically applies integrity, validation, and access-control policies. Valid transactions are recorded on the blockchain, providing a distributed and tamper-resistant history of data operations. Supporting information, such as user accounts and device metadata, is maintained in MySQL to

improve query performance and reduce unnecessary blockchain storage.

Lightweight cryptography is particularly important in the proposed architecture because IoT devices frequently operate under strict resource constraints. Algorithms and mechanisms should therefore be selected according to the required security level, computational capacity, memory availability, communication overhead, and energy budget of the target devices. Lightweight block ciphers and modern compact cryptographic mechanisms may be considered where appropriate, while established authenticated-encryption and public-key techniques can be used for stronger key-management requirements. The architecture is consequently designed around a hybrid security strategy rather than relying on a single cryptographic primitive. Blockchain provides distributed trust and auditability, cryptography protects confidentiality and integrity, and smart contracts automate policy enforcement. This combination creates a security architecture capable of supporting reliable weather-data management in IoT environments.

Development Tools

Features of Java

Java is a general-purpose, object-oriented programming language widely used for developing portable and scalable software applications. A major characteristic of Java is its platform-independent execution model. Java source code is compiled into bytecode, which is executed by the Java Virtual Machine (JVM). Consequently, an application can operate on different operating systems when an appropriate JVM is available. This property is useful for the proposed system because the application can be developed and tested in one environment and deployed on another without extensive modification. Java also provides automatic memory management, exception handling, multithreading, extensive standard libraries, and security-oriented APIs, making it suitable for implementing distributed and database-driven applications.

For the proposed system, Java provides the application-level foundation through which IoT data can be processed, cryptographic operations can be invoked, blockchain transactions can be managed, and database requests can be executed. Its object-oriented design also supports the modular implementation of system components such as weather-data processing, authentication, encryption, blockchain interaction, smart-contract communication, and dashboard services.

MySQL Database

MySQL is used as the relational database component for storing supporting system information. The database can maintain user accounts, roles, device metadata, access requests, system logs, configuration information, and references to blockchain transactions. Critical weather information is protected through the blockchain layer, while non-critical metadata remains in MySQL to enable efficient querying and application management.

The database design can include separate tables for users, devices, weather metadata, access requests, transaction references, and audit logs. Appropriate primary keys, foreign keys, indexes, constraints, and authentication mechanisms should be implemented to maintain consistency and security. JDBC provides the connection between the Java application and MySQL.

JSP Dashboard

The JSP dashboard serves as the primary web-based interface for authorized users and administrators. It provides facilities for authentication, weather-data visualization, device monitoring, transaction-status inspection, access-request management, and security-event monitoring. Weather parameters can be displayed through tables and graphical representations, allowing users to examine variations across locations and timestamps.

For administrators, the dashboard can provide functions for reviewing registered users, validating access requests, monitoring blockchain transactions, and examining audit logs. Role-based access control ensures that users can access only the functions and information permitted by their assigned privileges.

JDBC

JDBC provides the database connectivity layer between Java applications and MySQL. It enables the application to establish database connections, execute SQL statements, retrieve query results, and manage transactions. In the proposed system, JDBC is primarily used for supporting operations such as user registration, authentication-related data retrieval, device metadata management, access-request recording, and audit-log maintenance. Prepared statements and appropriate connection-management practices should be employed to reduce security risks and improve database performance.

Cryptographic Development Support

Cryptographic functionality is an essential part of the proposed security architecture. The application should use established and appropriately reviewed cryptographic libraries rather than implementing cryptographic primitives manually. The cryptographic layer is responsible for confidentiality, integrity verification, authentication, and secure key management. Depending on the final implementation and device constraints, lightweight symmetric encryption, authenticated encryption, hashing, digital signatures, and elliptic-curve-based key mechanisms can be integrated.

The cryptographic design should distinguish between **encryption**, which protects confidentiality,

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

and **hashing**, which provides integrity verification but does not encrypt information. Similarly, digital signatures provide authenticity and integrity but are not substitutes for encryption. Correct separation of these security functions improves the technical accuracy and security of the overall system.

Development Environment

The proposed system can be developed using an integrated development environment such as Eclipse or IntelliJ IDEA. Windows 10/11 can be used as the development operating system, while modern browsers such as Chrome, Firefox, or Edge can be used for testing the JSP dashboard. Apache Tomcat provides the web application runtime, and MySQL provides relational database services. The complete environment supports development, debugging, integration testing, security testing, and performance evaluation.

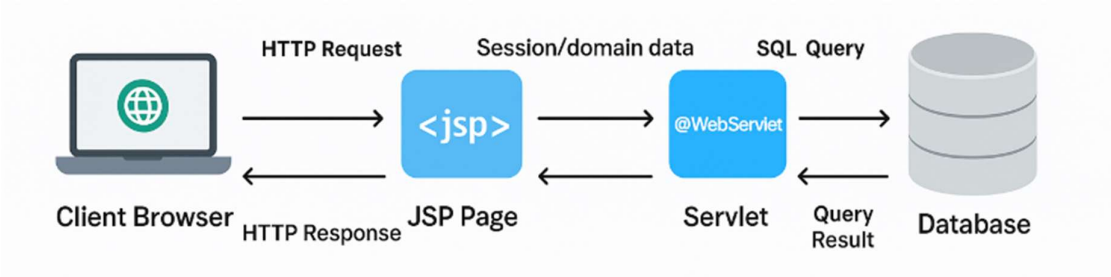

Figure 2: Development Tools

5. Implementation

The implementation phase converts the proposed design of the Secure IoT Weather Data Management System Using Blockchain and Lightweight Cryptography into an operational software solution. This phase integrates the IoT data collection module, cryptographic security mechanism, blockchain layer, smart contract engine, JSP-based dashboard, and MySQL database into a unified application. The implementation focuses on secure data acquisition, encryption, blockchain-based storage, policy validation, authorized data retrieval, and system monitoring. Each module is developed independently and subsequently integrated to ensure that the complete system operates according to the functional and security requirements defined during the design stage.

Implementation Approach

The implementation follows a modular architecture in which each component performs a specific responsibility while communicating with the other components through well-defined interfaces. IoT weather data such as temperature, humidity, atmospheric pressure, wind speed, rainfall, and location information is first collected from the device or simulated sensor environment. The received information is validated and associated with a device identifier and timestamp. Before transmission or permanent storage, the data is protected using the proposed lightweight cryptographic mechanism. The encrypted information is subsequently submitted to the blockchain layer, where transaction validation and integrity verification are performed. After successful verification, the transaction is recorded in the distributed ledger.

The smart contract engine provides automated security enforcement within the blockchain environment. It verifies predefined conditions, checks the integrity of incoming records, evaluates access requests, and identifies abnormal or inconsistent sensor information. Valid transactions are accepted for blockchain storage, whereas invalid or suspicious transactions are rejected or flagged for further inspection. This mechanism reduces dependence on manual intervention and provides a consistent approach to security-policy enforcement.

The JSP-based dashboard provides an interactive interface through which authorized users and administrators can access system functions. Users can authenticate themselves, view permitted weather records, search information according to location or timestamp, and monitor the status of blockchain transactions. Administrators can manage users, verify access requests, monitor security events, and review system logs. The MySQL database is used for supporting information such as user accounts, roles, device metadata, and operational logs, while critical weather information is protected through blockchain-based storage.

Implementation Workflow

The overall implementation follows a sequential security workflow. First, the IoT device collects weather information. Second, the received data is validated and assigned appropriate metadata. Third, the cryptographic module encrypts the data and generates the corresponding integrity information. Fourth, the protected transaction is submitted to the blockchain network. Fifth, the smart contract

verifies the transaction according to predefined security policies. Sixth, the validated transaction is recorded in the blockchain ledger. Finally, authorized users can retrieve and visualize the permitted information through the JSP dashboard. This workflow provides an integrated mechanism for secure collection, transmission, storage, validation, and visualization of IoT weather information.

6. Result and Software Testing

Software testing is performed to verify that the proposed Secure IoT Weather Data Management System Using Blockchain and Lightweight Cryptography functions correctly, securely, and efficiently. Testing identifies defects and confirms that the system meets its functional and non-functional requirements. Since the application combines IoT data collection, encryption, blockchain, smart contracts, database services, and a web dashboard, different levels of testing are required.

Test Methodology

The testing process begins with individual modules and gradually progresses to the complete system. Each component is tested separately and then integrated with other components. Functional, security, performance, integration, system, and acceptance testing are performed to verify the overall reliability and correctness of the application.

Unit Testing

Unit testing verifies individual modules independently. The authentication, weather-data validation, encryption, hashing, blockchain transaction, smart-contract, database, and dashboard components are tested using valid and invalid inputs. The objective is to confirm that each component produces the expected output.

Functional Testing

Functional testing determines whether the system performs the functions specified in the requirements. Weather data must be correctly accepted and validated, encryption must protect the data, blockchain must record valid transactions, and smart contracts must correctly approve or reject requests. The dashboard must also display authorized information properly.

System Testing

System testing evaluates the complete application as an integrated system. The process is tested from weather-data collection through encryption, smart-contract validation, blockchain storage, and dashboard retrieval. This testing confirms that all modules operate together correctly without data loss or unauthorized access.

Performance Testing

Performance testing measures the efficiency and responsiveness of the system. Parameters such as encryption time, transaction latency, database response time, dashboard response time, and throughput are examined. The objective is to ensure that security processing does not introduce unacceptable delays.

Integration Testing

Integration testing verifies communication between connected modules. Particular attention is given to the IoT, cryptographic, blockchain, smart-contract, database, and dashboard components. The purpose is to ensure that data is transferred correctly between modules without corruption or interface errors.

Security Testing

Security testing verifies that the system protects IoT weather information from unauthorized access and modification. Authentication and authorization mechanisms are tested, while encryption and integrity functions are examined for correct operation. Blockchain records are also checked to ensure that unauthorized changes can be detected.

Acceptance Testing

Acceptance testing determines whether the completed system satisfies user and project requirements. Users perform activities such as registration, login, weather-data submission, encryption, blockchain storage, data verification, and dashboard visualization. The system is accepted when these operations produce the expected results.

Test Plan

The test plan defines the test cases, inputs, expected results, observed results, and final status for each feature. Tests include valid and invalid login, weather-data submission, encryption and decryption, blockchain transactions, smart-contract decisions, unauthorized access, dashboard operations, and database management.

Testing Summary

The combined testing approach verifies the correctness, security, performance, and reliability of the proposed system. Unit, functional, integration, system, performance, security, and acceptance testing collectively help ensure that IoT weather data can be securely collected, protected, stored, validated, and accessed by authorized users.

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

Table 1. Functional and Module-Level Test Cases

Test ID	Module / Feature	Test Scenario	Test Input / Condition	Expected Result	Observed Result	Status
TC-01	User Authentication	Login with valid credentials	Registered username and correct password	User is authenticated and redirected to dashboard	User successfully logged in	Pass
TC-02	User Authentication	Login with invalid credentials	Incorrect username/password	Access is denied and error message is displayed	Invalid login rejected	Pass
TC-03	Weather Data Collection	Submit valid weather data	Temperature, humidity, pressure, wind speed and timestamp	Data is accepted and validated	Data successfully accepted	Pass
TC-04	Weather Data Validation	Submit incomplete data	Missing temperature or humidity value	System rejects incomplete record	Invalid record rejected	Pass
TC-05	Encryption	Encrypt weather information	Valid weather-data record	Data is converted into encrypted ciphertext	Encryption completed successfully	Pass
TC-06	Decryption	Decrypt authorized data	Valid encrypted record and key	Original weather data is recovered	Data recovered correctly	Pass
TC-07	Hash Generation	Generate data integrity hash	Valid weather-data record	Unique hash value is generated	Hash generated successfully	Pass
TC-08	Blockchain Transaction	Store verified weather record	Valid encrypted data and hash	Blockchain transaction is created successfully	Transaction recorded	Pass
TC-09	Smart Contract	Approve valid transaction	Authenticated user and valid transaction	Smart contract approves transaction	Transaction approved	Pass

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 45-60
ISSN: 3071-3358

TC -10	Smart Contract	Reject unauthorized transaction	Invalid user or invalid transaction	Smart contract rejects request	Request rejected	Pass
TC -11	Database	Store application metadata	Valid user/weather information	Record is stored without data corruption	Record stored correctly	Pass
TC -12	Dashboard	Display authorized weather data	Authenticated user requests data	Correct weather information is displayed	Data displayed correctly	Pass
TC -13	Dashboard	Access without authentication	Unauthenticated user attempts dashboard access	Access is denied	Unauthorized access blocked	Pass
TC -14	Data Retrieval	Retrieve blockchain-verified record	Valid transaction/hash	Correct record is retrieved and verified	Record retrieved successfully	Pass
TC -15	Data Integrity	Modify stored weather data	Altered data compared with original hash	Modification is detected	Integrity violation detected	Pass

Table 2. Security, Performance, Integration, and Acceptance Testing

Test ID	Testing Category	Test Scenario	Test Condition	Expected Result	Observed Result	Status
TC -16	Integration Testing	IoT to encryption module	Sensor sends weather data	Data reaches encryption module correctly	Data transferred correctly	Pass
TC -17	Integration Testing	Encryption to blockchain	Encrypted data with generated hash	Correct transaction is forwarded to blockchain	Transaction processed correctly	Pass
TC -18	Integration Testing	Blockchain to dashboard	Verified blockchain record requested	Authorized data is retrieved and displayed	Data displayed correctly	Pass
TC -19	Security Testing	Unauthorized data access	User without required privileges requests data	System denies access	Access denied	Pass

TC -20	Security Testing	Data tampering detection	Blockchain -related record is modified	Hash mismatch/tampering is detected	Tampering detected	Pass
TC -21	Security Testing	Password protection	Incorrect password attempts	Authentication mechanism prevents unauthorized login	Unauthorized login prevented	Pass
TC -22	Security Testing	Encrypted-data verification	Attempt to read protected weather data directly	Sensitive information remains unreadable	Encrypted data protected	Pass
TC -23	Performance Testing	Encryption processing time	Multiple weather records encrypted	Encryption completes within acceptable response time	Acceptable response observed	Pass
TC -24	Performance Testing	Blockchain transaction latency	Multiple valid transactions submitted	Transactions are processed without excessive delay	Transactions processed successfully	Pass
TC -25	Performance Testing	Database response	Weather records retrieved from database	Required records are returned promptly	Response time acceptable	Pass
TC -26	Performance Testing	Dashboard response	User requests weather information	Dashboard loads requested information without significant delay	Dashboard responded successfully	Pass
TC -27	Reliability Testing	Continuous weather-data submission	Repeated sensor data submissions	System processes records without unexpected failure	Continuous processing successful	Pass
TC -28	Error Handling	Invalid sensor data	Out-of-range or malformed values	System identifies and rejects invalid data	Invalid input handled correctly	Pass
TC -29	Acceptance Testing	Complete user workflow	Login → submit data → encrypt → blockchain → retrieve	Complete workflow operates successfully	Workflow completed successfully	Pass

TC -30	Acceptance Testing	End-to-end security verification	Authorized and unauthorized operations	Authorized operations succeed and unauthorized operations fail	Security requirements satisfied	Pass

7. Conclusion

The proposed Secure IoT Weather Data Management System Using Blockchain and Lightweight Cryptography provides an integrated framework for protecting weather information generated by IoT devices. The system combines IoT-based data collection, lightweight cryptographic protection, blockchain-based storage, smart contract validation, and web-based visualization to address important security and management requirements in distributed IoT environments. Weather parameters such as temperature, humidity, atmospheric pressure, wind speed, cloudiness, location, and timestamp are collected from IoT devices and processed before being transmitted to the secure storage layer.

Blockchain technology provides a decentralized and tamper-resistant storage mechanism for critical weather information. Instead of depending entirely on a centralized storage authority, the proposed system maintains verifiable records through a distributed ledger. Each validated transaction is associated with blockchain information such as a transaction identifier, timestamp, and cryptographic hash. This approach strengthens traceability and makes unauthorized modification of previously recorded information considerably more difficult. The blockchain layer therefore complements the cryptographic security mechanism by providing persistent integrity verification and auditability.

The Smart Contract Engine further strengthens the proposed architecture by automating data validation, security-policy enforcement, and anomaly identification. Smart contracts can evaluate incoming records according to predefined rules and determine whether the data satisfies the required conditions. Invalid or suspicious records can be flagged for further inspection, while compliant transactions can proceed through the storage workflow. This reduces dependence on manual verification and provides a consistent mechanism for enforcing access and validation policies.

The system also incorporates a JSP-based dashboard and MySQL database to improve usability and operational management. The dashboard enables authorized users and administrators to monitor weather records, transaction status, verification results, alerts, and graphical representations of collected parameters. MySQL is used for supporting information such as user credentials, roles, device metadata, and system logs, while critical weather information is protected through the blockchain layer. This hybrid storage arrangement provides a practical balance between accessibility, performance, and security.

From a system-design perspective, the integration of IoT, lightweight cryptography, blockchain, smart contracts, and web technologies creates a modular architecture that can be extended for different applications. The framework can support environments such as smart agriculture, environmental monitoring, disaster management, smart cities, and industrial monitoring. Its modular structure also allows additional weather parameters, IoT devices, analytical functions, and security policies to be incorporated without redesigning the complete system.

References

[1]. M. S. Uddin and M. D. Zainlabuddin, ‘‘Secure video processing and watermark embedding using Shamir secret rule,’’ Int. J. Artif. Intell. Comput. Electron., vol. 2, no. 1, pp. 25–31, Mar. 2026.

[2]. O. Peter, A. Pradhan, and C. Mbohwa, “Industrial Internet of Things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in emerging economies,” Procedia Computer Science, vol. 217, pp. 856–865, 2023.

[3]. V. K. B. Parasaram, V. T. Bathini, S. K. Nalluri, & A. H. Sannidhanam. (2026). A Sustainable AI-Enabled Predictive Maintenance Framework for Smart Industrial Systems Using Industrial IoT Data. 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS), 440–447. doi:10.1109/ICICDS70526.2026.11604878

[4]. H. Alshahrani, A. Khan, M. Rizwan, M. S. A. Reshan, A. Sulaiman, and A. Shaikh, "Intrusion detection framework for industrial Internet of Things using software defined network," Sustainability, vol. 15, no. 11, p. 9001, 2023.

[5]. Sannidhanam, A. H. (2021). Real-Time Claims Processing Using Event-Driven Architectures. International Journal of Technology, Management and Humanities, 7(01), 36–50. doi:10.21590/07.01.02

[6]. B. Babayigit and M. Abubaker, "Industrial Internet of Things: A review of improvements over traditional SCADA systems for industrial automation," IEEE Systems Journal, vol. 18, no. 1, pp. 120–133, 2024.

[7]. J. Singh, I. S. Ahuja, H. Singh, and A. Singh, "Application of Quality 4.0 (Q4.0) and Industrial Internet of Things (IIoT) in agricultural manufacturing industry," AgriEngineering, vol. 5, no. 1, pp. 537–565, 2023.

[8]. Anjani Haritha Sannidhanam. (2024). Prompt Engineering Patterns for Reliable LLM Outputs in Production Environments. Journal of Multidisciplinary Knowledge, 4(1), 29–39.

[9]. A. Sasikumar, S. Vairavasundaram, K. Kotecha, V. Indragandhi, L. Ravi, G. Selvachandran, and A. Abraham, "Blockchain-based trust mechanism for digital twin empowered industrial Internet of Things," Future Generation Computer Systems, vol. 141, pp. 16–27, 2023.

[10]. R. Kumar and N. Agrawal, "Analysis of multi-dimensional industrial IoT (IIoT) data in edge–fog–cloud based architectural frameworks: A survey on current state and research challenges," Journal of Industrial Information Integration, vol. 35, Art. no. 100504, 2023.

[11]. Sannidhanam, A. H., & Alladi, S. (2017). Cloud-Based Healthcare CRM Systems for Improved Patient Engagement. International Journal of Technology, Management and Humanities, 3(01), 32–44. doi:10.21590/ijtmh.3.03.4

[12]. D. Kumar, P. Pawar, H. Gonaygunta, and S. Singh, "Impact of federated learning on industrial IoT—A review," International Journal of Advanced Research in Computer and Communication Engineering, vol. 13, no. 1, pp. 1–12, 2023.

[13]. V.-T. Truong, D.-B. Ha, A. Nayyar, M. Bilal, and D. Kwak, "Performance analysis and optimization of multiple IIoT devices radio frequency energy harvesting NOMA mobile edge computing networks," Alexandria Engineering Journal, vol. 79, pp. 1–20, 2023.

[14]. A. Aljuhani, P. Kumar, R. Alanazi, T. Albalawi, O. Taouali, A. K. M. N. Islam, N. Kumar, and M. Alazab, "A deep learning integrated blockchain framework for securing industrial IoT," IEEE Internet of Things Journal, vol. 11, no. 5, pp. 7817–7827, 2024.

[15]. Performance Analysis of Wireless Sensor Networks for Smart Monitoring Applications. (2016). International Journal of Humanities and Information Technology, 1(04), 10–29. doi:10.21590/ijhit.01.04.04

[16]. H. Alasmary, "RDAF-IIoT: Reliable device-access framework for the industrial Internet of Things," Mathematics, vol. 11, no. 12, p. 2710, 2023.

[17]. J. V. Arputharaj and S. K. Pal, "Transforming Industry 5.0: Real-time monitoring and decision making with IIOT," in Sustainability in

Industry 5.0. Boca Raton, FL, USA: CRC Press, 2024, pp. 76–106.

[18]. Sannidhanam, A. H. (2020). Comparative Analysis of Cloud Computing Architectures for Enterprise Applications. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 12(02), 169–180. doi:10.18090//samriddhi.v12i02.16

[19]. D. Berestov, O. Kurchenko, L. Zubyk, S. Kulibaba, and N. Mazur, "Assessment of weather risks for agriculture using big data and industrial Internet of Things technologies," in Proceedings of Cybersecurity Providing Information and Telecommunication Systems, 2023, pp. 1–13.

[20]. M. Kumar, G. K. Walia, H. Shingare, S. Singh, and S. S. Gill, "AI-based sustainable and intelligent offloading framework for IIoT in collaborative cloud-fog environments," IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 1414–1422, 2024.

[21]. Sannidhanam, A. H. (2023). Zero-Downtime AI Model Updates in Real-Time Inference Systems. International Journal of Engineering Science & Humanities, 13(2), 70–78.

[22]. K. S. Hawaou, V. C. Kamla, S. Yassa, O. Romain, J. E. N. Mboula, and L. Bitjoka, "Industry 4.0 and industrial workflow scheduling: A survey," Journal of Industrial Information Integration, vol. 38, Art. no. 100546, 2024.

[23]. O. T. Sanchez, D. Raposo, A. Rodrigues, F. Boavida, R. Marculescu, K. Chen, and J. Sá Silva, "An IIoT-based approach to the integrated management of machinery in the construction industry," IEEE Access, vol. 11, pp. 6331–6350, 2023.

[24]. M. D. Zainlabuddin and N. Sharma, ''Security enhancement in data propagation for wireless network,'' Rev. GEINTEC-Gestão, Inovação e Tecnologias, vol. 11, no. 4, pp. 4110–4119, Aug. 2021.

[25]. Anjani Haritha Sannidhanam. (2026). LLM-Driven Workflow Optimization: Intelligent Routing in Asynchronous Distributed Systems. International Journal of AI and Machine Learning, 1(2), 23–33.

[26]. Shaik Abdul Waheed, Mohammed Sulaiman, Mirza Awaiz Baig, Dr. Mohammed Abdul Bari," Intelligent Conversational Agent for Seamless User Interaction Using NLP and Dialog flow",International Journal of Information Technology and Computer Engineering, SSN 2347–3657 ,Volume 13, Special Issue 2s, 2025, Page -91-98.

[27]. Vishwanath Nikhil , Dr. Mohammed Abdul Bari ," Real Time Powerlifting Form Assessment using Yolov5 and Mediapipe", International Journal of Information Technology and Computer Engineering, SSN 2347–3657, Volume 13, Special Issue 2s, 2025, Pages 574-584