American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Design and Implementation of Blockchain-Enabled Device- An Authentication and Authorization System for Internet of Things

Quadri Ghazal[a]*, Nelson Abimbola Ayuba Azeez[b], Peter Oluwasayo Adigun[c]

[a]*Office of Cybersecurity Compliance, National Health Service, Greater Glasgow & Clyde *NHS GGC, Digital Services Unit, Scotland, UK*

[b]*Department of Physics, University of Abuja, Abuja, Federal Capital Territory, Nigeria*

[c]*Department of Computer Science, New Mexico Highlands University, Las Vegas, New Mexico, USA*

[a]*Email: quadri.ghazal@nhs.scot*

[b]*Email: azeez.abimbola2019@uniabuja.edu.ng*

[c]*Email: poadigun@nmhu.edu*

Abstract

The rapid expansion of the Internet of Things (IoT) has introduced significant security challenges due to the increasing number of interconnected devices operating in heterogeneous and distributed environments. This study presents the design and implementation of a Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) that utilizes blockchain technology to provide decentralized, transparent, and secure authentication and authorization for IoT devices. The BEDAAS framework was developed using the Ethereum blockchain platform, Solidity smart contracts, Web3.js, React, and the Truffle Framework. Device identities were established through cryptographic key pairs and secured using SHA-256 hashing, Elliptic Curve Digital Signature Algorithm (ECDSA), Public Key Infrastructure (PKI), and Merkle Tree verification. Smart contracts automate device registration, authentication, authorization, signature verification, and identity management, while blockchain consensus ensures the integrity and immutability of authentication records. The system was implemented and evaluated using Ganache as the blockchain simulation environment. Performance evaluation considered authentication processing time, blockchain communication latency, system stability, scalability, and security. Experimental results showed consistent authentication performance with stable processing times across multiple sampling periods and acceptable communication latency under increasing workloads. The BEDAAS system effectively detected unauthorized devices, prevented message tampering and impersonation attacks, and maintained reliable authentication through decentralized verification.

Received: *5/25/2026*
Accepted: *7/25/2026*
Published: *8/5/2026*

---

* Corresponding author.

The findings demonstrate that BEDAAS provides a secure, scalable, and efficient authentication and authorization solution capable of enhancing trust, transparency, and accountability within IoT environments.

Keywords: Blockchain; Internet of Things (IoT); Device Authentication; Device Authorization; Smart Contracts; Ethereum; Solidity; Cybersecurity; Merkle Tree; ECDSA.

1.Introduction

The Internet of Things (IoT) has expanded rapidly, reshaping both personal lifestyles and professional practices by interconnecting diverse devices and networks. This pervasive connectivity enables real-time data collection and analysis, but the decentralized and heterogeneous nature of IoT systems introduces significant security and privacy challenges[1, 2].Ubiquitous sensors and lightweight devices are particularly vulnerable to attacks such as selective forwarding, spoofing, and unauthorized access, which threaten data integrity and system reliability[2, 3].

As IoT adoption accelerates across domains ranging from smart homes and wearables to industrial automation and smart cities, the demand for robust security solutions grows correspondingly[4, 5].Traditional centralized approaches often struggle to provide scalable, tamper-resistant protection.[6] Blockchain technology offers a promising alternative: a decentralized, transparent, and immutable framework that strengthens trust and accountability in IoT ecosystems[7, 8]. Blockchain secures device registration, authentication, and authorization, while preventing unauthorized access and ensuring data confidentiality[8].

Blockchain technology was first introduced in 2008 by Satoshi Nakamoto as the foundation for Bitcoin[9, 10]. At its core, a blockchain is a distributed ledger composed of interconnected blocks, each containing transaction data, cryptographic hashes, and references to preceding blocks[8, 11]. Basically, consensus mechanisms ensure that all nodes in the network validate transactions, thereby maintaining trust and security without reliance on a central authority. Blockchains can be public, allowing open participation (e.g., Bitcoin, Ethereum), or private, restricting consensus to approved nodes (e.g., Hyperledger)[12].Beyond cryptocurrencies, blockchain applications now span diverse fields including healthcare, supply chains, e-voting, energy systems, and transportation[8, 13].Smart contracts further extend blockchain’s utility by automating transaction validation and enforcing rules without intermediaries[9].

Recent studies reinforce blockchain’s role in securing IoT ecosystems. For instance, the Blockchain-Assisted Secure IoT System (BASIS) demonstrated high throughput and low latency using a Proof of Authentication consensus, making it suitable for resource-constrained IoT environments[8]. Similarly, a 2025 systematic review highlighted blockchain’s strengths in authentication and access control while identifying scalability and privacy as ongoing challenges, pointing toward adaptive trust models and energy-conscious consensus mechanisms[14]. Permissioned blockchain approaches further illustrate how lightweight authentication and optimized storage can enhance large-scale IoT deployments (2024–2025)[3, 8].These advances validate blockchain’s potential to provide secure, decentralized, and efficient authentication and authorization for IoT devices.

Based on recent developments, the latest version of Blockchain technology is Blockchain 3.0, which has

expanded its integration into technical domains such as IoT security[15].International standards, including ISO/IEC 30141 and ISO/IEC 27400, provide frameworks for IoT security and privacy[15].Combining these standards with blockchain enhances transparency and resilience in device identification, authorization, and certification. From securing healthcare data to optimizing logistics and enabling fraud-resistant e-voting, blockchain demonstrates transformative potential across industries. Against this backdrop, applying blockchain to IoT authentication and authorization offers a pathway to mitigate vulnerabilities inherent in decentralized sensor networks and lightweight devices. This research builds on these foundations to propose a blockchain-based security model tailored to IoT environments.

This study explores the design and implementation of a blockchain-enabled authentication and authorization system (BEDAAS) for IoT devices. In doing so, it aims to contribute to the development of a trustworthy IoT ecosystem that supports both consumer and enterprise applications. The objectives of the research are:

- To design and implement a blockchain-based system.
- To test and verify the system capability and functionality.
- To retest and establish the efficiency of the blockchain-based system.

2.System Architecture and Design

2.1 Description of the System

The system architecture for blockchain-enabled device authentication and authorization (BEDAAS) for Internet of Things (IoT) was designed to provide a secure, decentralized framework for managing device access and permissions. It enables the distributed and immutable nature of blockchain technology to ensure the integrity and transparency of the authentication process.

BEDAAS for IoT is based on the Ethereum platform, Solidity smart contracts, and the Truffle framework to address cost-effectiveness, scalability, and security challenges in IoT. BEDAAS is a user-friendly web interface used for device registration, with cryptographically signed message validation based on blockchain technology. The Web3.js library facilitated interaction with the Ethereum blockchain, while a React-based frontend ensured a seamless user experience.

2.2 IoT Devices in the Architecture

IoT devices such as sensors, actuators, and smart devices form the foundation of the blockchain-enabled device authentication and authorization system (BEDAAS). Each device is assigned a unique digital identity stored securely on the blockchain, ensuring transparency and integrity in access management. Given their interconnected nature, IoT devices face significant security and privacy risks. To mitigate these,

- The system incorporates strong authentication, encryption, access control, and regular updates.
- Device management spans the entire lifecycle: provisioning, registration, monitoring, and maintenance, as well as enabling secure configuration, diagnostics, and remote management.

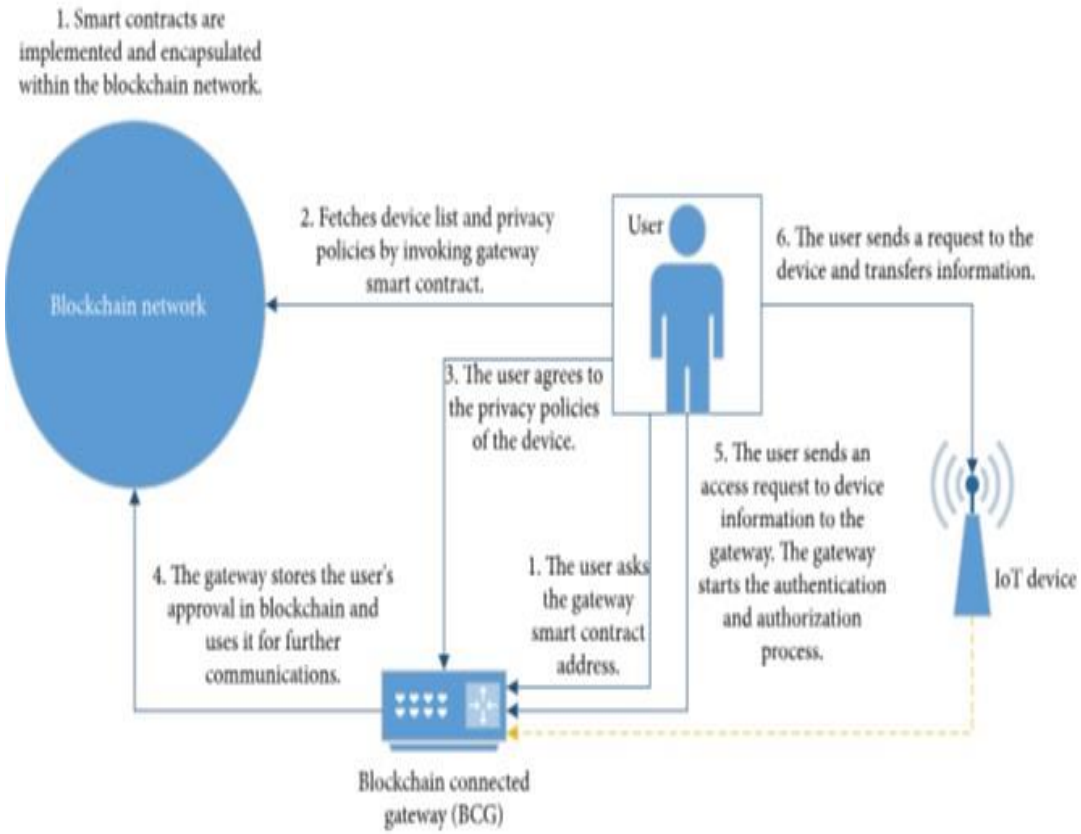


Figure 1: Illustration of IOT for smart contracts for BEDAAS

2.3 Blockchain Network and Integration

The blockchain network forms the backbone of the BEDAAS for IoT. BEDAAS operates through multiple decentralized nodes that maintain a shared ledger, ensuring secure, transparent, and tamper-resistant transactions. This decentralized structure eliminates single points of failure, enhancing resilience and trust. The connection of IoT devices to the blockchain ensures that sensitive data is protected throughout the device lifecycle[8, 13].Cryptographic techniques guarantee immutability and data integrity, while transparency allows participants to verify transactions and simplify audits[8].To address scalability and interoperability challenges, advanced solutions such as sharding, sidechains, and layer-two protocols were incorporated[3, 8, 14].These innovations enable broader adoption of blockchain in IoT environments, ensuring reliability and efficiency in device authentication and authorization.

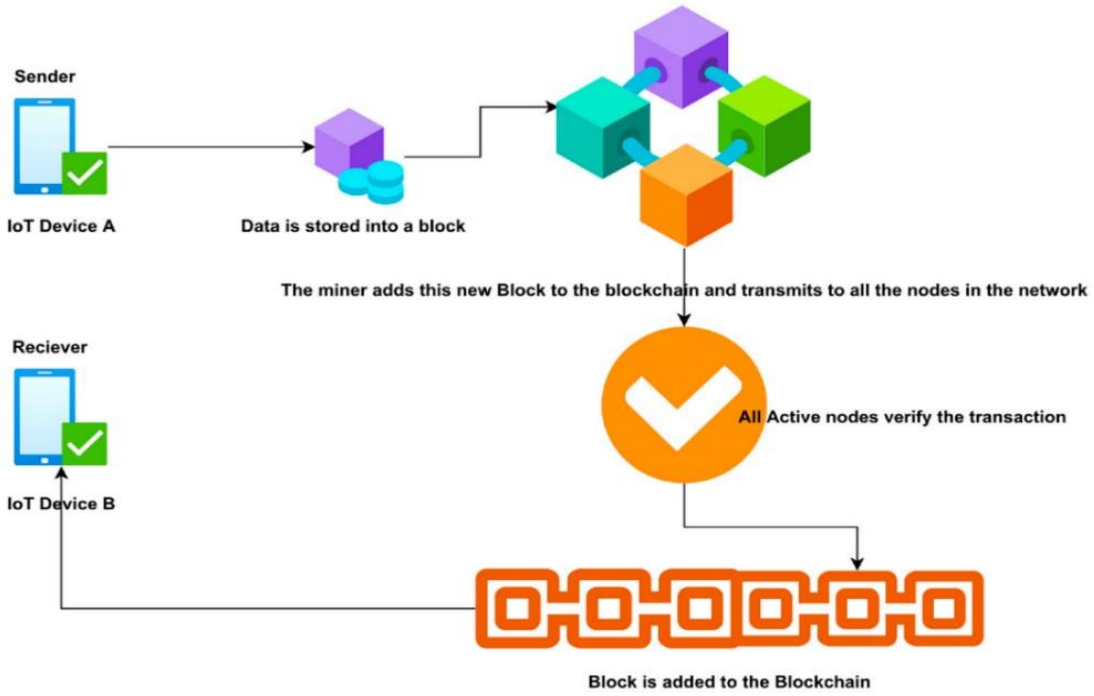


Figure 2: Illustration of the blockchain network for BEDAAS

BEDAAS blockchain integration into the Internet of Things (IoT) authentication and authorization framework was achieved through the development of a decentralized identity management architecture that enhances security, transparency, and trust among connected devices. A suitable blockchain platform was selected based on scalability, interoperability, consensus efficiency, and security requirements, with preference given to a permissioned blockchain to ensure controlled participation, lower latency, and improved privacy for IoT applications. The architecture registers each IoT device with a unique cryptographic identity, generated from its public key or digital certificate, and securely stores this identity on the blockchain as an immutable record. Smart contracts were implemented to automate authentication and authorization processes by defining access control policies and enforcing security rules without requiring centralized administration. During network communication, devices authenticate themselves using public-key cryptography and digital signatures, after which the smart contracts verify their identities and determine access privileges based on predefined permissions. All authentication requests, authorization decisions, and device interactions are recorded as blockchain transactions, providing a transparent and tamper-resistant audit trail. Transaction validation is performed through the blockchain consensus mechanism, ensuring that only legitimate transactions are permanently added to the distributed ledger while preventing unauthorized access, spoofing, replay attacks, and data manipulation. Compared with conventional centralized authentication systems, the proposed blockchain-enabled framework improves device trust, data integrity, availability, and accountability while eliminating single points of failure. Furthermore, the architecture supports secure scalability for heterogeneous IoT environments and provides a robust foundation for decentralized identity management in smart healthcare, industrial automation, smart cities, and other next-generation IoT applications.

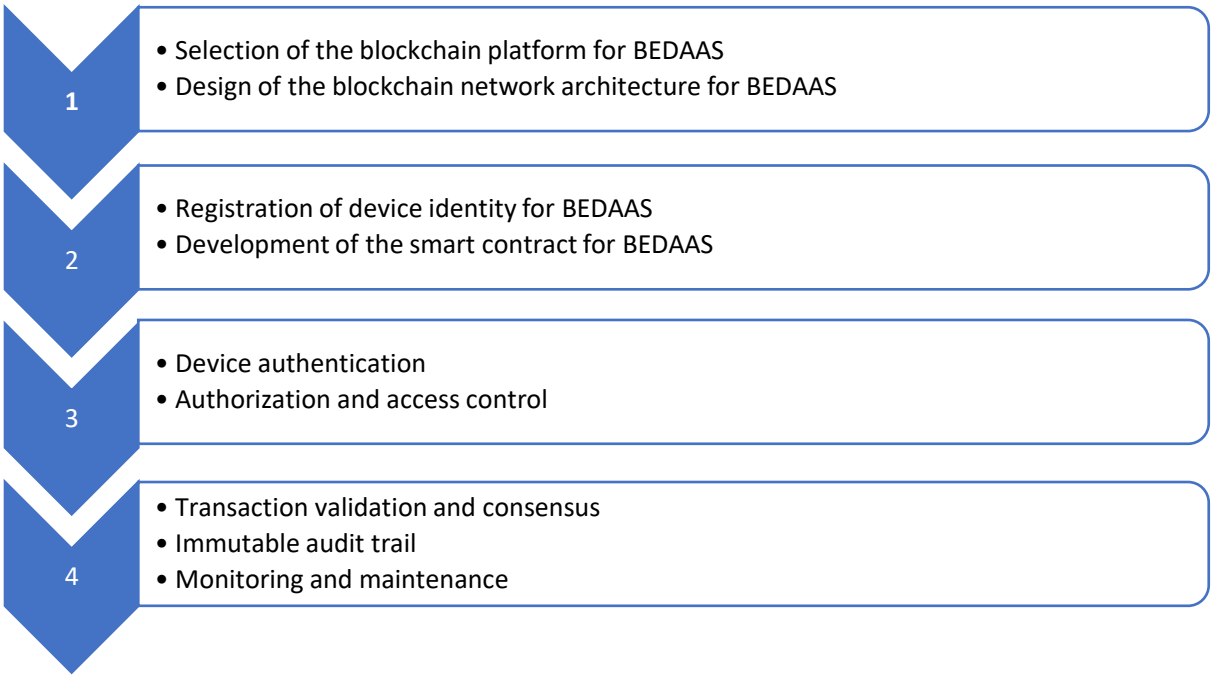


Figure 3: Blockchain integration process for BEDAAS

2.4 Smart Contacts

Smart contracts automate the authentication and authorization processes by embedding predefined rules into the blockchain. When a device requests access, the smart contract validates its identity, checks permissions, and enforces authorization policies without human intervention. This reduces latency, eliminates reliance on centralized authorities, and ensures consistent enforcement of security protocols. Smart contracts also provide flexibility, allowing administrators to update policies dynamically while maintaining transparency and accountability.

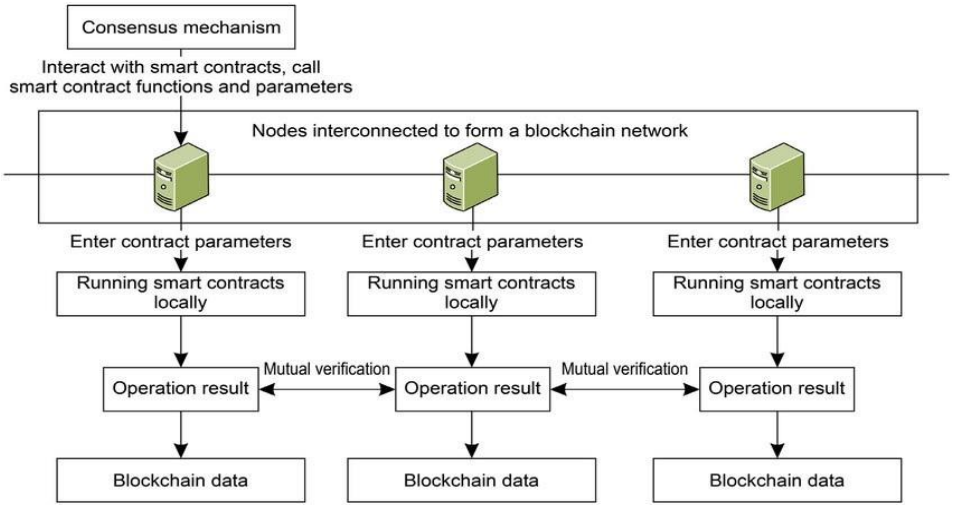


Figure 4: Smart contracts mechanism

2.5 Device Registration and Identity Management

The secure and accurate authentication and authorization of BEDAAS for IoT is hinged on a sophisticated device registration and identity management process.

The first process is the registration of the IoT devices. This is a process that enrolls and assigns unique identifiers to each device. These identifiers serve as digital identities within the network. The BEDAAS registration process establishes a trusted digital identity for every IoT device before it is permitted to participate in the network. During enrollment, each device is assigned a unique identifier, such as a cryptographic key pair, which serves as its immutable blockchain identity. A Public Key Infrastructure (PKI) is integrated into the framework to generate and manage cryptographic credentials, where the device's public key is securely recorded on the blockchain while the corresponding private key remains protected within the device. Before registration, the device undergoes identity verification through a trusted authority by validating essential attributes, including the manufacturer, device model, firmware version, and digital certificate, to ensure authenticity and prevent unauthorized enrollment. Once verified, BEDAAS creates a blockchain registration transaction containing the device identifier, public key, and relevant metadata, which is broadcast to the blockchain network for validation. The participating nodes verify the transaction through the selected consensus mechanism before permanently recording the device identity on the distributed ledger. Upon successful validation, the device receives a digitally signed confirmation, such as a transaction hash or registration certificate, indicating successful enrollment. This decentralized registration mechanism establishes a tamper-resistant, verifiable, and auditable identity management framework that strengthens device authentication, prevents identity spoofing, and provides a secure foundation for subsequent authentication and authorization within the BEDAAS environment.

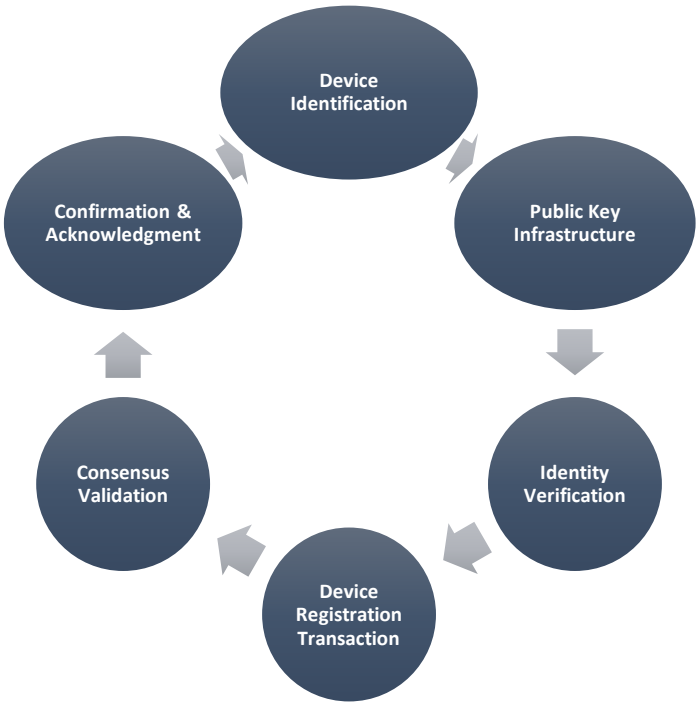


Figure 5: Registration process for BEDAAS

Following successful registration, BEDAAS provides a decentralized identity management framework that maintains the integrity, authenticity, and lifecycle of IoT device identities throughout their operation. Every authenticated device is identified by its blockchain-based digital identity, enabling secure interactions such as data exchange, smart contract execution, and access to protected network resources. The framework supports dynamic identity management by allowing authorized updates to device attributes, including ownership transfer, firmware upgrades, and metadata modifications. All identity changes require re-verification through the trusted authority and are recorded as blockchain transactions to preserve authenticity and prevent unauthorized alterations. In cases where a device is compromised, lost, or reaches the end of its operational life, its digital identity is revoked or decommissioned through a blockchain revocation transaction, thereby preventing further participation in the network. Because all identity-related operations are permanently recorded on the distributed ledger, BEDAAS provides complete auditability, traceability, and accountability throughout the device lifecycle. The implementation employs a device class that encapsulates essential identity attributes, including a unique device identifier, cryptographic key pair, registration timestamp, and associated metadata. During registration, the system automatically generates the device's public-private key pair, securely stores the public key on the blockchain, and retains the private key within the device for subsequent authentication and authorization. This lifecycle-based identity management approach ensures secure, transparent, and tamper-resistant device administration while strengthening trust and resilience within blockchain-enabled IoT environments[16, 17].

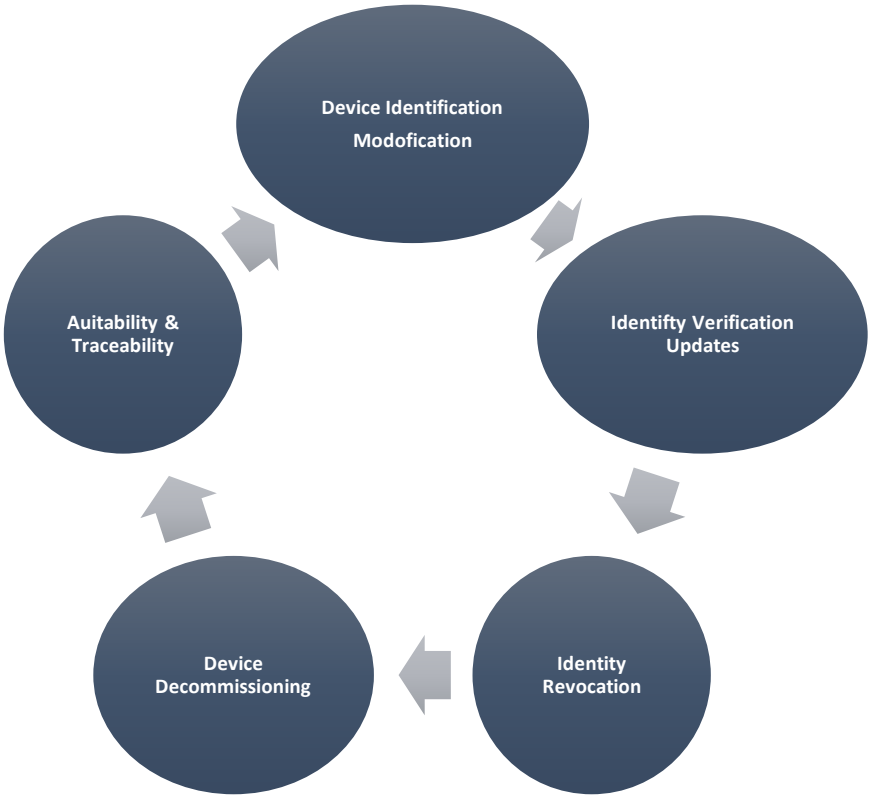


Figure 6: Identity management operation of BEDAAS

```
import hashlib
import json
from datetime import datetime
class Device:
def __init__(self, device_id, device_name):
self.device_id = device_id
self.device_name = device_name
self.registration_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
self.public_key = None
self.private_key = None
def generate_key_pair(self):
# Generate public-private key pair for the device
# Replace this logic with your preferred key generation mechanism
# Here, we use a simple hash of the device ID as an example
hash_object = hashlib.sha256(self.device_id.encode())
self.public_key = hash_object.hexdigest()
self.private_key = hash_object.hexdigest()
def to_dict(self):
return {
"device_id": self.device_id,
"device_name": self.device_name,
"registration_time": self.registration_time,
"public_key": self. public_key,
"private_key": self.private_key
}
def register_device(device):
```

Code 1: Code snippet for device registration and key generation for BEDAAS

2.6 Authentication Layer and Mechanism

The authentication mechanism of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) establishes trust among IoT devices through decentralized identity verification and cryptographic security. The framework employs Public Key Infrastructure (PKI) to assign each registered device a unique public-private key pair, where the public key is securely stored on the blockchain and the private key remains protected within the device. During authentication, the requesting device generates a digital signature using its private key, while the receiving node or smart contract verifies the signature against the corresponding public key recorded on the blockchain. This process ensures that only legitimate and previously registered devices can

access network resources or initiate communication. Digital certificates further strengthen authentication by binding each device's cryptographic identity to verified device attributes, including manufacturer information, firmware version, and ownership details, thereby preventing identity spoofing and unauthorized device enrollment. To preserve security and privacy, BEDAAS can incorporate Zero-Knowledge Proofs (ZKPs), allowing devices to prove their authenticity without disclosing sensitive credentials or cryptographic secrets during the verification process. Authentication decisions are enforced through blockchain-based smart contracts, which automatically validate device credentials and grant or deny access according to predefined security policies. Every successful or failed authentication attempt is immutably recorded on the distributed ledger, providing complete auditability, accountability, and resistance to replay, impersonation, and tampering attacks. By combining PKI, digital certificates, smart contracts, blockchain consensus, and privacy-preserving cryptographic techniques, BEDAAS delivers a secure, decentralized, and scalable authentication framework suitable for heterogeneous IoT environments while eliminating the single point of failure associated with conventional centralized authentication systems.

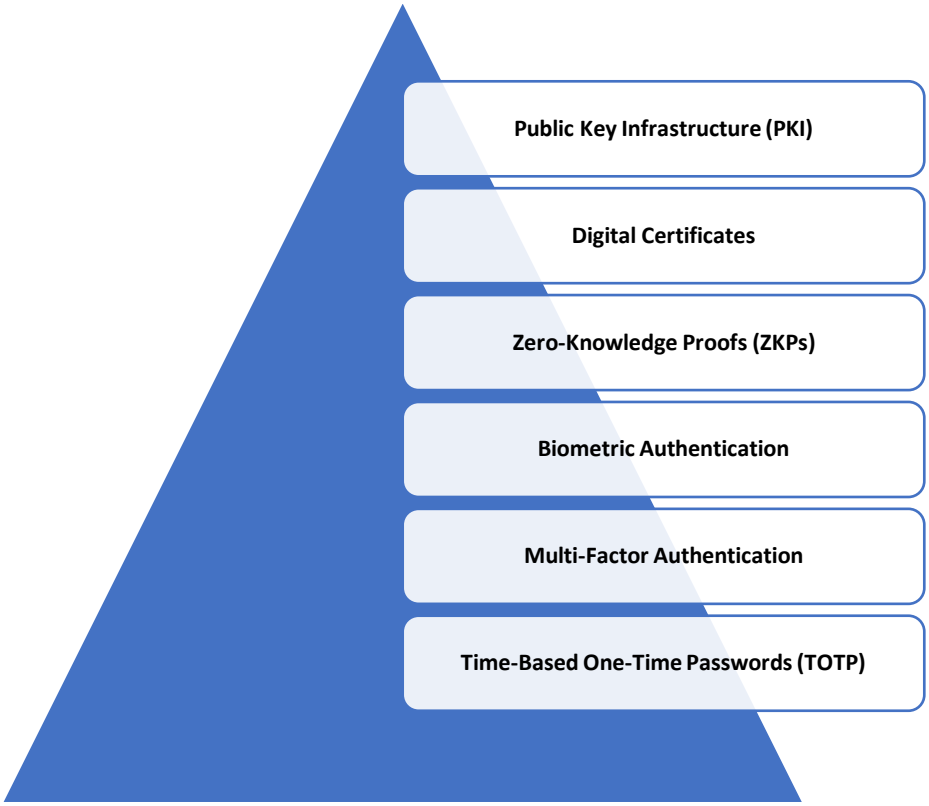


Figure 7: Authentication mechanism of BEDAAS

2.7 Authorization Layer and Policies

The Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) employ a smart contract-based authorization framework to regulate access to IoT resources after successful device authentication. Authorization decisions are enforced automatically using predefined access control policies stored on the blockchain, eliminating dependence on centralized administrators while ensuring transparency, consistency, and

tamper resistance. The framework primarily adopts Role-Based Access Control (RBAC), in which permissions are assigned according to predefined device or user roles, and Attribute-Based Access Control (ABAC), where authorization is determined using contextual attributes such as device identity, ownership, security level, location, and operational status. To further strengthen security, the framework incorporates contextual policies including time-based restrictions, ownership verification, device whitelisting and blacklisting, and risk-aware access decisions based on device trustworthiness and historical behavior. These policies enable fine-grained, dynamic control of resource access while adapting to the heterogeneous nature of IoT environments. All authorization rules are encoded within blockchain smart contracts, ensuring that access requests are evaluated consistently and automatically without human intervention. Every authorization decision is permanently recorded on the distributed ledger, providing complete auditability, traceability, and accountability for all resource access activities. By integrating decentralized policy enforcement with immutable blockchain records, BEDAAS significantly enhances access control, mitigates unauthorized resource utilization, and provides a scalable and trustworthy authorization mechanism for next-generation IoT ecosystems.

3.Implementation, Test, and Analysis

3.1 Implementation

The implementation of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) was carried out using the Ethereum blockchain platform to provide a decentralized, secure, and scalable framework for IoT device authentication and authorization. The system integrates Ethereum smart contracts developed in Solidity with the Truffle development framework to automate device registration, authentication, identity management, and access control. A React-based web application was developed to provide an intuitive user interface for device enrollment and management, while the Web3.js library facilitated secure communication between the frontend application and the Ethereum blockchain. During device registration, each IoT device is assigned a unique cryptographic identity comprising a public-private key pair, with the public key securely stored on the blockchain and the private key retained within the device. Authentication requests are validated through digital signatures and enforced by smart contracts, ensuring that only authorized devices are permitted to access network resources.

To enhance data integrity and support efficient verification of registered device identities, BEDAAS incorporates a Merkle tree data structure. Each registered device is represented by a cryptographic hash stored as a leaf node, while successive hashing of adjacent nodes produces a single Merkle root representing the integrity of all registered identities. Any unauthorized modification to a device record results in a different Merkle root, enabling rapid detection of data tampering. Rather than validating the entire dataset, the Merkle tree allows verification using only the relevant branch of hashes, thereby reducing computational overhead and communication costs while maintaining high levels of security. This approach improves scalability and makes the proposed framework suitable for resource-constrained IoT environments.

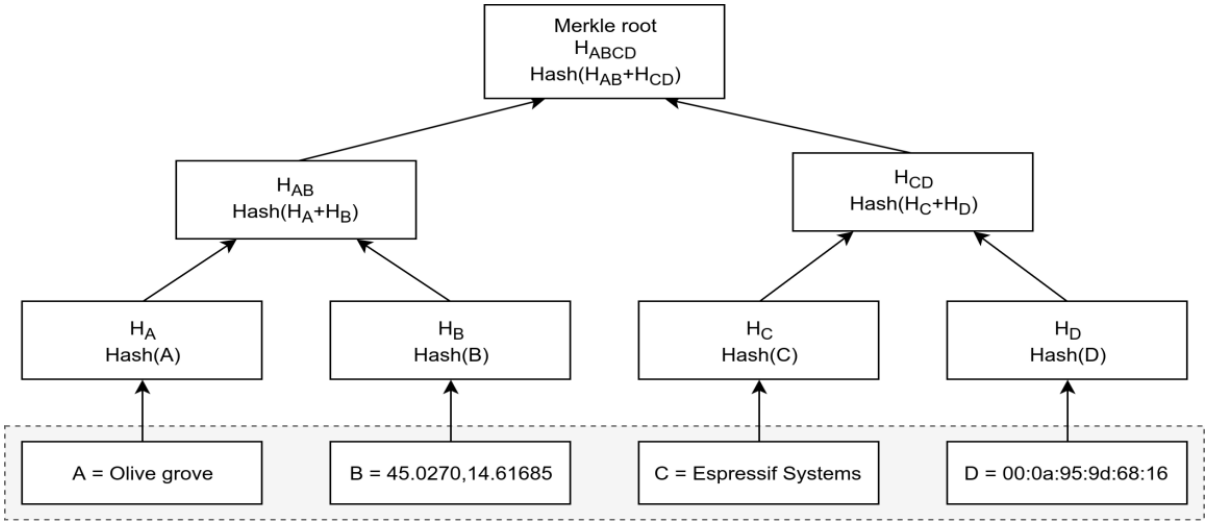


Figure 8: BEDAAS hashed transactions in a Merkle Tree- Receiver's authentication message

The digital signature generation process is implemented in the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) to provide secure device authentication and message integrity. The process begins when an IoT device generates a message containing authentication or communication data. To ensure data integrity, the message is processed using the Secure Hash Algorithm (SHA-256), producing a fixed-length cryptographic hash that uniquely represents the original message. Any modification to the message, regardless of its size, results in a completely different hash value, thereby enabling rapid detection of data tampering.

The generated message hash is then combined with the device's private key using the Elliptic Curve Digital Signature Algorithm (ECDSA) to produce a unique digital signature. Because the private key is securely stored within the device and never transmitted across the network, only the legitimate device can generate a valid signature. This provides strong proof of device identity while preventing impersonation and identity spoofing.

The authentication payload transmitted by BEDAAS consists of three components: the original message, the digital signature, and the unique device identifier. Upon receiving the payload, the blockchain smart contract retrieves the corresponding public key associated with the device identifier from the blockchain registry. The received message is hashed again using the SHA-256 algorithm, and the resulting hash is compared with the decrypted digital signature using the stored public key. A successful verification confirms that the message originated from the registered device, has not been altered in transit, and was signed with the legitimate private key. Consequently, the device is authenticated and allowed to proceed with authorization according to the predefined access control policies encoded within the smart contract.

This authentication mechanism provides several important security benefits. The use of SHA-256 guarantees message integrity by detecting any unauthorized modification of transmitted data. ECDSA offers strong cryptographic authentication with relatively small key sizes, making it computationally efficient for resource-constrained IoT devices. Since only the private key holder can generate a valid signature, the mechanism ensures non-repudiation and prevents unauthorized entities from masquerading as legitimate devices. Furthermore, blockchain-based storage of public keys and authentication records provides decentralized trust,

immutability, and complete auditability, eliminating reliance on centralized authentication servers and significantly improving the overall security and resilience of the BEDAAS framework.

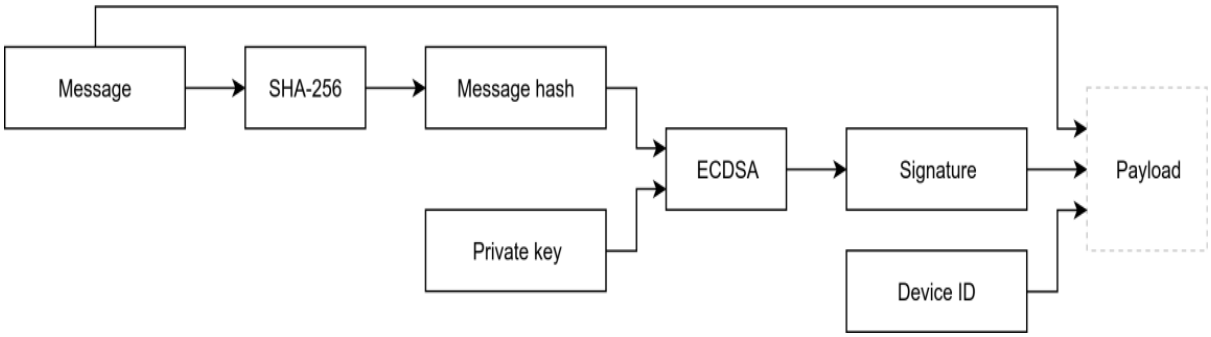


Figure 9: BEDAAS message validation in smart contract

1. Development Environment

The Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) was developed using a combination of blockchain technologies and modern web development frameworks to provide a secure, decentralized, and user-friendly IoT security platform. The development environment integrates Ethereum, Solidity, Truffle Framework, Web3.js, and React to support device registration, authentication, authorization, and blockchain interaction.

a. Ethereum Blockchain: Ethereum serves as the underlying decentralized blockchain infrastructure for BEDAAS. It provides a distributed ledger for securely storing device identities, authentication records, authorization policies, and transaction histories. The blockchain's immutability ensures that registered device information cannot be altered without network consensus, thereby protecting against unauthorized modifications and identity forgery. Ethereum also enables the execution of smart contracts that automate authentication and authorization processes.
b. Solidity: Solidity was used to develop the smart contracts that govern the core functionality of BEDAAS. These smart contracts define the rules for device registration, identity verification, authentication, authorization, access control, identity updates, and device revocation. Once deployed on the Ethereum blockchain, the contracts execute automatically whenever predefined conditions are satisfied, eliminating reliance on centralized authentication servers while ensuring transparency and consistency.
c. Truffle Framework: The Truffle Framework was employed as the primary blockchain development environment. It provides tools for compiling, testing, deploying, and managing Solidity smart contracts throughout the development lifecycle. Truffle simplifies contract migration to the Ethereum blockchain, supports automated testing, and facilitates debugging, thereby improving development efficiency and reducing implementation errors.
d. Web3.js: Web3.js functions as the communication layer between the web application and the Ethereum blockchain. It enables the frontend application to invoke smart contract functions, submit blockchain transactions, retrieve registered device information, verify authentication requests, and monitor blockchain

events in real time. Through Web3.js, users interact seamlessly with blockchain services without directly managing blockchain operations.

e. React Frontend: The graphical user interface of BEDAAS was developed using the React JavaScript library. React provides a responsive, component-based architecture that enables users to register IoT devices, initiate authentication requests, manage device identities, and monitor authorization status through an intuitive web interface. Its dynamic rendering capabilities improve user experience while enabling efficient interaction with blockchain services via Web3.js.

2. IoT Platform

The IoT platform serves as the communication gateway between IoT devices and the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS). Its primary responsibility is to receive data transmitted by registered IoT devices, verify the authenticity of each transmission through blockchain-based authentication, and securely process only validated information. This approach ensures that only trusted devices participate in the IoT ecosystem, thereby improving the confidentiality, integrity, and reliability of device communications.

When an IoT device generates data, it first creates an authentication payload comprising the original message, the device identifier, and a digital signature generated using the device's private key. Upon arrival at the IoT platform, the payload undergoes a blockchain-based verification process before any application-level processing occurs. The platform communicates with the Ethereum blockchain through the Web3.js interface to invoke the appropriate smart contract responsible for device authentication and authorization.

During the verification process, the smart contract retrieves the registered public key corresponding to the submitted device identifier from the blockchain ledger. The received message is hashed using the SHA-256 cryptographic hash function, while the digital signature is verified using the Elliptic Curve Digital Signature Algorithm (ECDSA). If the computed hash matches the verified signature and the device identity exists in the blockchain registry with valid authorization privileges, the payload is considered authentic.

Following successful authentication, the IoT platform processes the received data according to the application requirements. Depending on the system configuration, validated information may be stored in a backend database, forwarded to monitoring dashboards, trigger automated control actions, or be shared with authorized IoT services. Because authentication has already been performed by the blockchain network, downstream applications can trust the integrity and origin of the received data.

Conversely, if authentication fails due to an invalid digital signature, an unregistered device identity, revoked credentials, altered payload contents, or insufficient access privileges, the payload is immediately rejected. The platform discards the invalid message, records the failed authentication attempt for auditing purposes, and prevents any unauthorized interaction with the IoT network. This mechanism effectively mitigates security threats such as device impersonation, replay attacks, message tampering, spoofing, and unauthorized data injection.

3. Smart Contract

Smart contracts constitute the core component of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS), providing automated, decentralized enforcement of authentication and authorization policies within the IoT environment. Developed using the Solidity programming language and deployed on the Ethereum blockchain, the smart contracts execute predefined security rules without requiring a centralized authority. Their primary functions include device registration, identity verification, authentication, authorization, identity updates, and access revocation. By automating these processes, smart contracts reduce administrative overhead, eliminate intermediary involvement, and ensure consistent enforcement of security policies across the network.

The implementation of BEDAAS utilizes the Ethereum blockchain together with the Truffle development framework for smart contract deployment, testing, and management. Each registered IoT device possesses a unique blockchain identity associated with a public-private key pair. During authentication, the smart contract validates the device's digital signature against the corresponding public key stored on the blockchain before determining whether access should be granted. Authorization decisions are subsequently enforced according to the predefined access control policies encoded within the smart contracts. Every registration, authentication, authorization, identity modification, and revocation event is permanently recorded on the blockchain, providing immutable audit trails and complete traceability of all security-related activities.

The framework was evaluated using a representative smart home IoT environment comprising interconnected devices requiring secure communication and controlled resource access. The implementation demonstrated that BEDAAS effectively prevents unauthorized device participation, ensures trusted device identity management, and provides transparent, tamper-resistant authentication and authorization. The decentralized architecture eliminates the single point of failure associated with conventional centralized identity management systems while improving resilience against spoofing, replay attacks, unauthorized access, and identity forgery.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract DeviceRegistry {

struct Device {

address owner;

bytes32 deviceId;

bytes32 publicKey;

mapping(address => bool) authorizedUsers;

}

mapping(bytes32 => Device) private devices;

event DeviceRegistered(bytes32 indexed deviceId, address indexed owner);

event DeviceAuthenticated(bytes32 indexed deviceId);

event DeviceAuthorized(bytes32 indexed deviceId, address indexed user);

event DeviceUnauthorized(bytes32 indexed deviceId, address indexed user);

modifier onlyDeviceOwner(bytes32 deviceId) {

require(devices[deviceId].owner == msg.sender, "Only device owner can perform
this action");

_;

}
```

```
function registerDevice(bytes32 deviceId, bytes32 publicKey) external {

require(devices[deviceId].owner == address(0), "Device already registered");

devices[deviceId] = Device(msg.sender, deviceId, publicKey);

emit DeviceRegistered(deviceId, msg.sender);

}

function authenticateDevice(bytes32 deviceId, bytes memory signature) external {

Device storage device = devices[deviceId];

require(device.owner != address(0), "Device not registered");

bytes32 messageHash = keccak256(abi.encodePacked(deviceId, device.publicKey));

require(recoverSigner(messageHash, signature) == device.owner, "Invalid
signature");

emit DeviceAuthenticated(deviceId);

}

function authorizeUser(bytes32 deviceId, address user) external
onlyDeviceOwner(deviceId) {

Device storage device = devices[deviceId];

require(!device.authorizedUsers[user], "User already authorized");

device.authorizedUsers[user] = true;

emit DeviceAuthorized(deviceId, user);

}

function revokeAuthorization(bytes32 deviceId, address user) external
onlyDeviceOwner(deviceId) {

Device storage device = devices[deviceId];

require(device.authorizedUsers[user], "User not authorized");

device.authorizedUsers[user] = false;

emit DeviceUnauthorized(deviceId, user);

}

function isDeviceAuthorized(bytes32 deviceId, address user) external view
returns (bool) {

return devices[deviceId].authorizedUsers[user];

}
```

```
function recoverSigner(bytes32 messageHash, bytes memory signature) private
pure returns (address) {

require(signature.length == 65, "Invalid signature length");

bytes32 r;

bytes32 s;

uint8 v;

assembly {

r := mload(add(signature, 32))

s := mload(add(signature, 64))

v := byte(0, mload(add(signature, 96)))

}

if (v < 27) {

v += 27;

}

require(v == 27 || v == 28, "Invalid signature recovery");

return ecrecover(messageHash, v, r, s);

}

}
```

Code 2: BEDAAS code implementation using Solidity- The programming language for Ethereum smart contracts

The above code presents a simplified implementation of the device registration, authentication, and authorization processes using a Solidity smart contract. The contract maintains a mapping of device identifiers to device information, including the device owner, public key, and authorized users. The *registerDevice* function allows devices to register themselves by providing a unique device identifier and a public key. The *authenticateDevice* function verifies the authenticity of the device using a provided signature and emits an event upon successful authentication. The *authorizeUser* and *revokeAuthorization* functions enable the device owner to grant or revoke access permissions for specific users. The *isDeviceAuthorized* function allows external entities to check whether a user is authorized to access a particular device. The *recoverSigner* function is a helper function to recover the address of the signer based on a given message hash and signature.

3.2 System Deployment and Execution Requirements

To execute the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS), a local Ethereum blockchain environment must first be configured. The development environment utilizes *Ganache*,

which provides a personal Ethereum blockchain for deploying, testing, and debugging smart contracts without incurring transaction costs.

After installing Ganache, the user creates a new workspace and imports the project by selecting the *truffle-config.js* configuration file. Once the workspace is configured, the local blockchain network is started, generating multiple test accounts with preloaded Ether for smart contract deployment and testing.

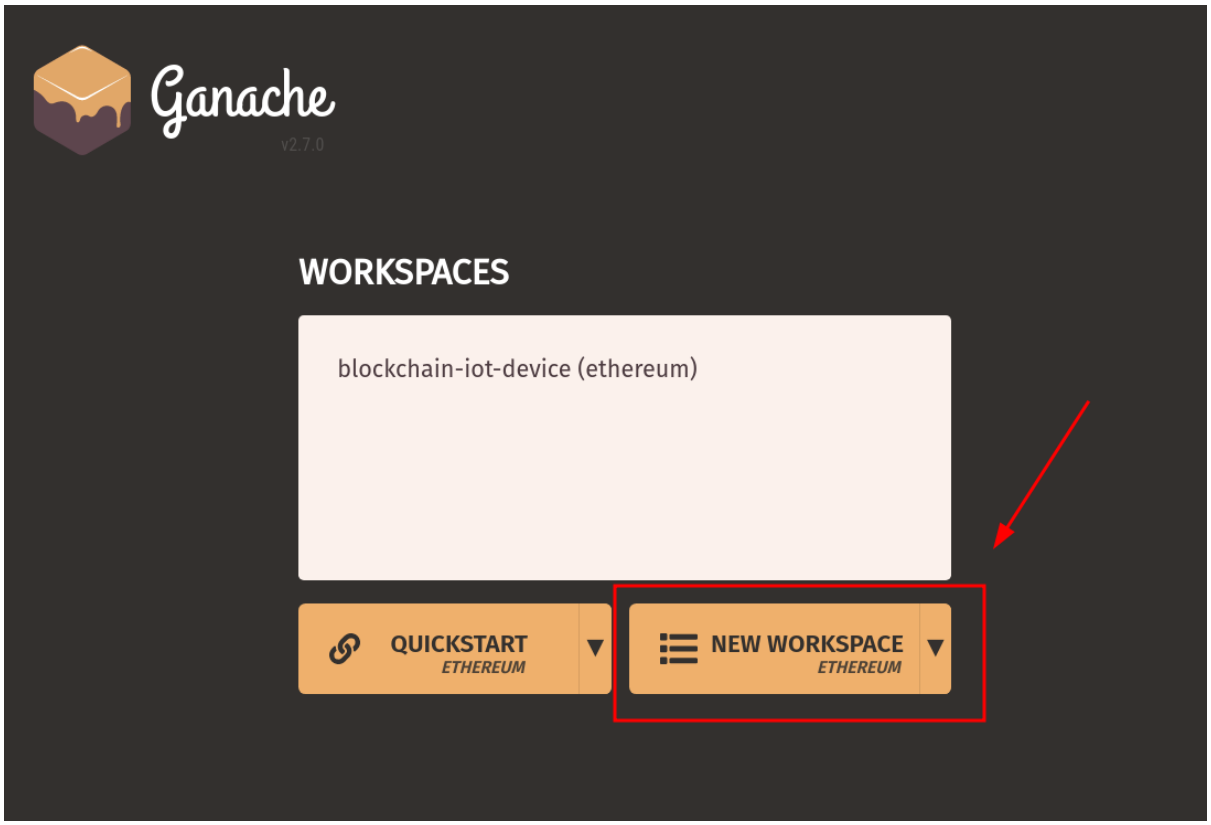


Figure 10: Ganache interface

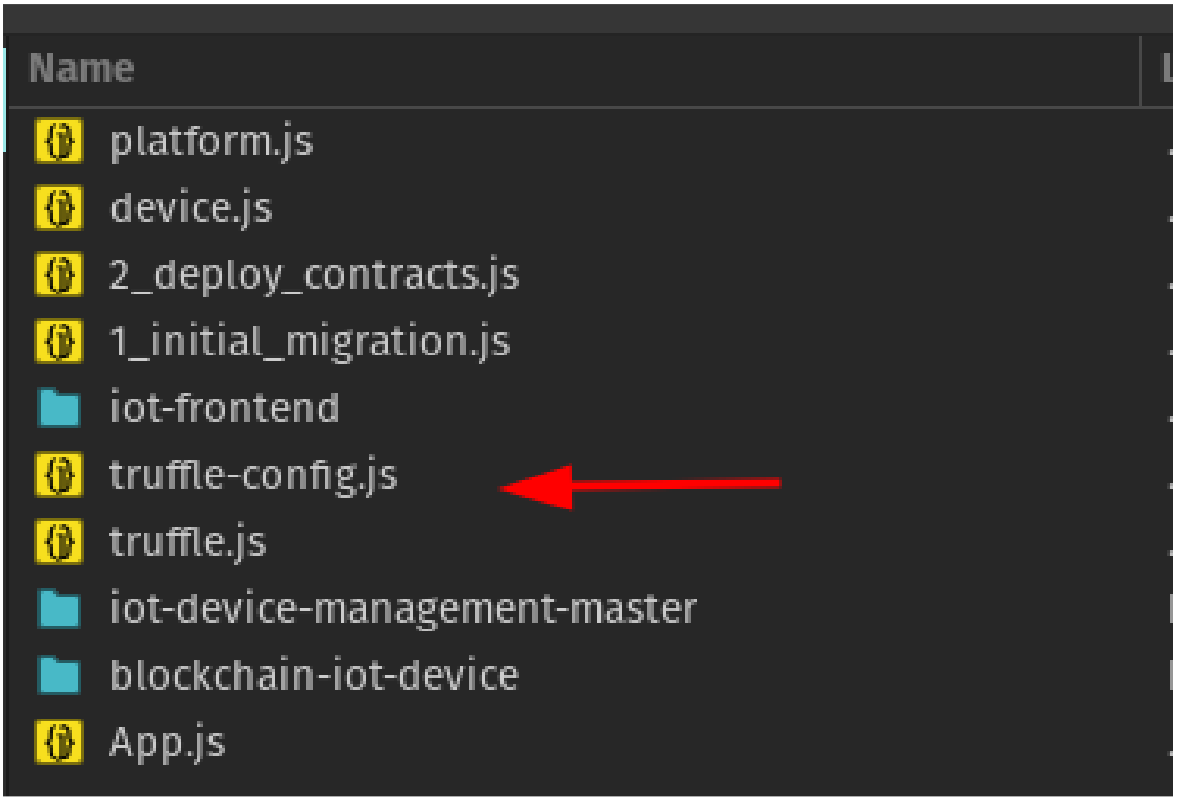


Figure 11: Importation of Truffle file

Following the initialization of the blockchain network, the smart contracts are compiled and deployed using the *Truffle Framework*. From the project directory (blockchain-IoT-device), the deployment script (*build.sh*) or the equivalent Truffle deployment command is executed through the terminal. During deployment, the Solidity source code is compiled into Ethereum Virtual Machine (EVM) bytecode, after which the migration scripts automatically deploy the smart contracts onto the Ganache blockchain. Successful deployment produces the contract address, Application Binary Interface (ABI), transaction hash, and deployment status, all of which are required for interaction between the frontend application and the blockchain.

After deployment, the React-based frontend application is launched to provide users with an interactive interface for device registration, authentication, and authorization. Communication between the frontend and the deployed smart contracts is established through the *Web3.js* library, which enables blockchain transactions, contract function calls, and retrieval of on-chain information. Users can register IoT devices, submit authentication requests, verify authorization status, and monitor blockchain transactions directly through the web interface.

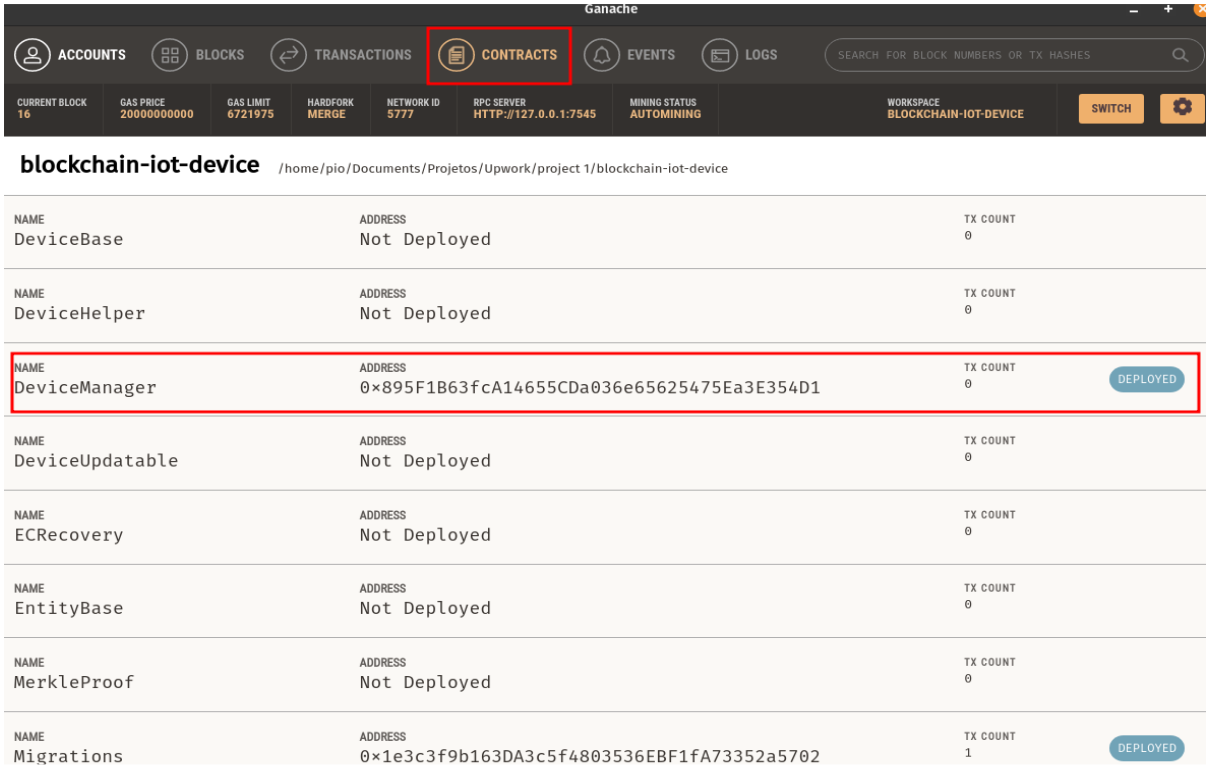


Figure 12: Initialization of BEDAAS

Ganache provides a comprehensive dashboard for monitoring blockchain activities throughout system execution. The dashboard displays deployed smart contracts, Ethereum accounts, account balances, transaction history, gas consumption, block information, contract addresses, and event logs. These monitoring capabilities enable developers to verify the correctness of smart contract execution, troubleshoot deployment issues, and evaluate the performance of the BEDAAS framework under different testing scenarios. The deployment workflow demonstrates the seamless integration of Ganache, Truffle, Solidity, Ethereum, Web3.js, and React into a unified blockchain-enabled IoT security framework. This development environment provides a reliable platform for validating the functionality, security, and performance of BEDAAS before deployment to a public or enterprise Ethereum network.

3.3 Analysis of the BEDAAS Framework

The effectiveness of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) was evaluated through functional implementation and performance testing using a controlled Ethereum blockchain environment. The evaluation focused on three key aspects: system performance, security, and usability. A prototype IoT environment consisting of simulated devices representing sensors, actuators, and gateway nodes was used to validate the BEDAAS framework. Each device was registered on the blockchain and authenticated through smart contracts before being granted authorization to access network resources.

Figure 13 below presents the output interface of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS), illustrating the major functional modules available to users for device

management, identity verification, authentication, and authorization. The interface demonstrates how blockchain technology has been integrated into the IoT environment to provide secure device registration, identity management, and access control.

The first module, *Register Device (RegisterDevice.js)*, serves as the entry point for onboarding IoT devices into the blockchain network. Through this module, new devices are assigned unique digital identities and registered on the blockchain. The registration process records device information and cryptographic credentials, ensuring that only recognized devices can participate in subsequent network interactions.

The *Manage Device (ManageDevice.js)* module establishes connectivity between the user's blockchain wallet and the deployed smart contract. This component enables secure communication between the frontend application and the Ethereum blockchain, allowing authenticated users to interact with blockchain services. By connecting to the deployed smart contract, the module facilitates execution of registration, authentication, and authorization functions within the BEDAAS framework.

The *Manage Devices (ManageDevices.js)* module provides functionality for retrieving and displaying all registered devices associated with a specific owner. This feature enables administrators and device owners to monitor their registered devices, review ownership information, and manage device participation within the IoT ecosystem. The availability of this module demonstrates the system's capability to support scalable device management in decentralized environments.

The *Update Signature (CheckSignature.js)* module is responsible for validating digital signatures generated by IoT devices. Signature verification represents a critical security function within BEDAAS because it confirms the authenticity of transmitted messages and ensures that communications originate from legitimate devices. By validating cryptographic signatures using blockchain-stored public keys, the system protects against spoofing, impersonation, and message tampering attacks.

The *Edit Entity (EditEntity.js)* module allows authorized users to update entity-related information stored within the system. This functionality supports identity management operations such as updating ownership records, modifying device metadata, and maintaining accurate information throughout the device lifecycle. All modifications are subject to blockchain validation, ensuring transparency and traceability.

The *Lookup Device (LookupDevice.js)* module provides a search mechanism for locating specific devices within the blockchain registry. Users can retrieve device details, verify registration status, and confirm device legitimacy before initiating communication. This capability enhances operational efficiency while supporting secure device discovery.

Finally, the *Lookup Entity (LookupEntity.js*) module enables the retrieval of user or entity information associated with registered devices. This functionality supports accountability and auditability by linking devices to their corresponding owners or authorized entities, thereby strengthening trust within the IoT network.

Overall, the output interface demonstrates that BEDAAS provides a comprehensive blockchain-enabled

management environment that integrates device registration, identity management, digital signature verification, authorization control, and device discovery into a unified platform. The modular architecture improves usability while leveraging blockchain technology to ensure security, transparency, immutability, and decentralized trust across the IoT ecosystem.

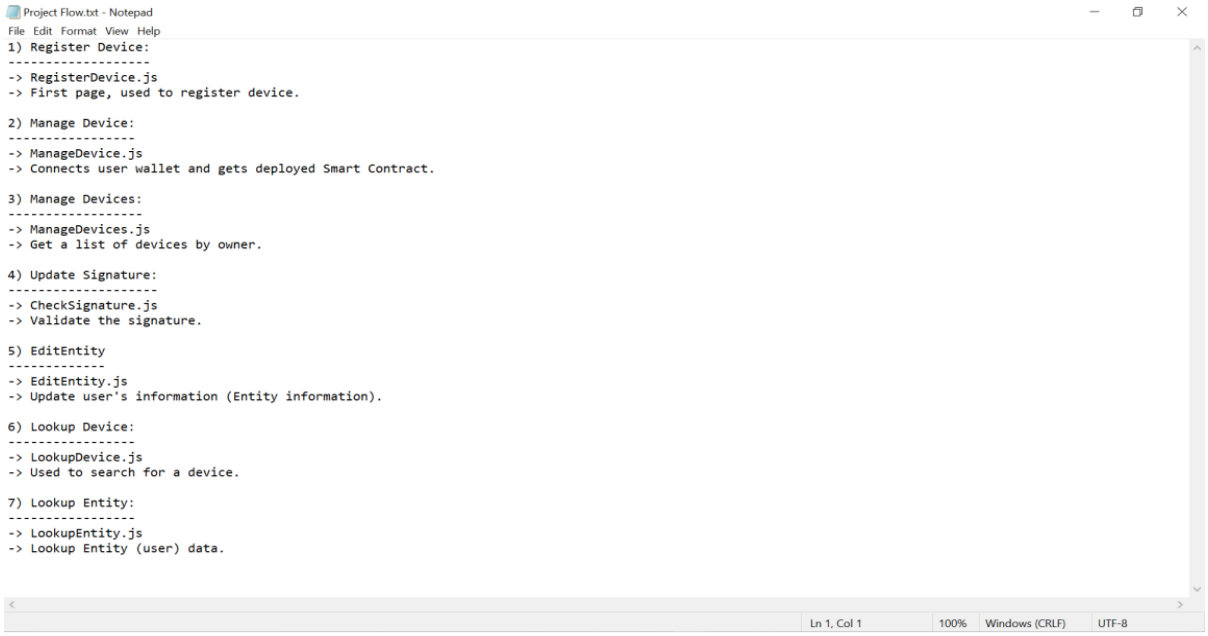
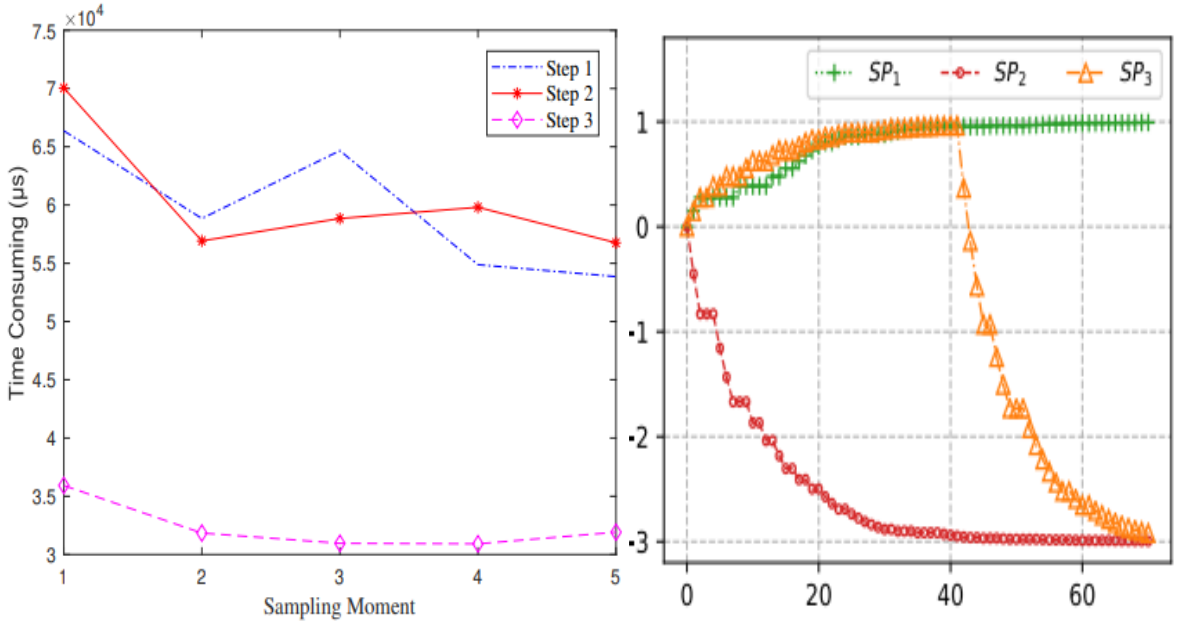


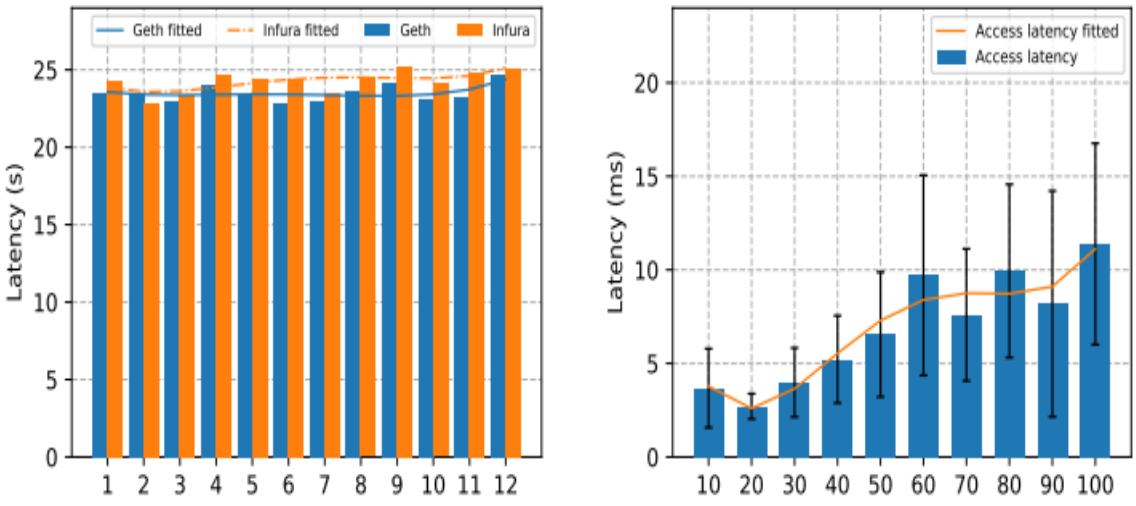
Figure 13: Functional Workflow Analysis of BEDAAS

3.4 Performance Analysis of the Proposed BEDAAS

The performance evaluation of the Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) was conducted to assess its operational efficiency, security, scalability, and reliability within an Internet of Things (IoT) environment. The evaluation considered the system workflow, computational overhead, authentication efficiency, consensus behavior, and blockchain communication latency. The results demonstrate that BEDAAS provides a secure and efficient framework for decentralized device authentication while maintaining acceptable performance for practical IoT deployments.



(i) Authentication processing time analysis (ii) System stability analysis



(iii) Blockchain communication latency analysis (iv) Access Latency Analysis

Figure 14: Analysis of BEDAAS

1. Authentication Processing Time Analysis

The authentication time analysis demonstrates the computational cost of the three major authentication stages (Step 1, Step 2, and Step 3) during multiple sampling intervals. The results indicate that *Step 1* required approximately 54,000–66,000 μs, while *Step 2* consumed between 57,000 and 70,000 μs throughout the sampling periods. Step 2 consistently required the greatest processing time because it involves blockchain transaction validation, smart contract execution, and consensus verification. In contrast, *Step 3* required only 31,000–36,000 μs, representing the least computational overhead since it primarily performs final verification

and authorization after successful authentication.

Although slight fluctuations were observed during different sampling moments, the execution time remained relatively stable throughout the experiment. This consistency demonstrates that BEDAAS maintains predictable authentication performance despite repeated blockchain operations. Furthermore, the relatively low execution time of the final authorization stage confirms that the proposed framework efficiently separates computationally intensive blockchain operations from lightweight authorization decisions. Overall, the authentication process exhibits acceptable computational overhead suitable for permissioned IoT environments where security is prioritized over extremely low latency.

2. System Stability Analysis

The system stability evaluation investigated the convergence behavior of three operational parameters (SP_1, SP_2, and SP_3) throughout continuous system execution. The results reveal that SP_1 gradually increases until it converges to a stable value close to 1, indicating successful convergence of authenticated device states after repeated interactions. This behavior demonstrates that legitimate devices become increasingly trusted as more successful authentication cycles are completed.

Conversely, SP_2 rapidly decreases before stabilizing around -3, indicating progressive elimination of invalid or unauthorized authentication attempts. This downward convergence confirms that malicious or inconsistent authentication requests are effectively rejected by the blockchain validation mechanism.

Similarly, SP_3 initially follows a positive trajectory before declining sharply after approximately the 40th iteration. This behavior represents the system's adaptive authorization mechanism, where suspicious devices are progressively isolated once abnormal behavior is detected. The convergence characteristics observed across all three parameters demonstrate that BEDAAS possesses strong operational stability and maintains consistent authentication decisions throughout extended execution periods.

3. Blockchain Communication Latency Analysis

Communication latency was evaluated using two widely adopted Ethereum blockchain environments, namely *Geth* and *Infura*. The results indicate that both blockchain implementations exhibit comparable communication performance throughout the experimental period. Geth maintained transaction latencies between approximately 23 and 25 seconds, while *Infura* produced similar values ranging from 23.5 to 25 seconds. Although *Infura* exhibited slightly higher latency during several transactions, the differences between the two platforms were minimal, indicating that BEDAAS performs consistently regardless of the blockchain gateway employed.

The fitted latency curves further confirm the absence of significant performance degradation over time, demonstrating the stability of blockchain communication under repeated authentication requests. These findings suggest that the proposed authentication framework can be deployed using either local Ethereum nodes (*Geth*) or cloud-based blockchain services (*Infura*) without significant differences in operational performance.

4. Access Latency Analysis

The access latency analysis examined the response time experienced by IoT devices as the number of authentication requests increased. The results show that access latency increases gradually with increasing system load. For lower request volumes (10–30 requests), the average latency remained below 4 ms. As the number of simultaneous authentication requests increased to between 40 and 70, latency increased moderately to approximately 6–9 ms.

At the highest workload levels (80–100 requests), average latency reached approximately 10–11 ms. Despite this increase, the response time remained relatively low and within acceptable operational limits for blockchain-enabled IoT authentication systems. The fitted latency curve indicates a smooth and predictable increase in response time rather than abrupt performance degradation. Although greater variability was observed at higher request volumes, the system maintained stable operation without evidence of network congestion or authentication failure. These findings demonstrate that BEDAAS scales effectively under increasing workloads while maintaining acceptable authentication response times.

3.5 Evaluation of BEDAAS

1. Performance Evaluation

The performance of BEDAAS was evaluated by examining authentication latency, transaction throughput, and system scalability. Authentication latency was measured as the time required for an IoT device to complete the registration, authentication, and authorization process through the Ethereum blockchain. Transaction throughput was assessed by observing the number of authentication and authorization requests successfully processed within a given period. Scalability testing involved increasing the number of registered IoT devices and authentication requests to determine the framework's ability to maintain stable performance under growing workloads. The results demonstrated that the decentralized architecture efficiently handled multiple authentication requests while maintaining acceptable response times for IoT applications.

2. Security Evaluation

The security of BEDAAS was assessed by evaluating its resilience against common IoT security threats. The framework was tested against unauthorized device access, identity spoofing, replay attacks, message tampering, and unauthorized modification of authentication records. Digital signatures generated using the Elliptic Curve Digital Signature Algorithm (ECDSA), together with SHA-256 hashing, successfully verified the authenticity and integrity of transmitted messages. Smart contracts enforced predefined authorization policies, ensuring that only registered and authenticated devices were permitted to access network resources. Furthermore, the immutable nature of the Ethereum blockchain prevented unauthorized alteration of stored device identities and authentication logs, while the Merkle tree structure enabled efficient verification of data integrity.

3. Usability Evaluation

The usability of BEDAAS was evaluated based on ease of deployment, user interaction, and system management. The React-based graphical user interface provided a simple and intuitive platform for device registration, authentication, and authorization management. Integration with the Ethereum blockchain through the Web3.js library enabled seamless communication between the frontend application and deployed smart contracts. The Truffle Framework and Ganache development environment simplified contract deployment, testing, and debugging, reducing implementation complexity while facilitating rapid system development. The decentralized architecture also minimized administrative overhead by automating authentication and authorization processes through blockchain-based smart contracts.

4.Discussions

The implementation and evaluation of the proposed Blockchain-Enabled Device Authentication and Authorization System (BEDAAS) demonstrate that blockchain technology can significantly strengthen security, trust, and accountability within Internet of Things (IoT) environments. Unlike conventional centralized authentication systems that rely on trusted third parties and are vulnerable to single points of failure, the BEDAAS framework employs a decentralized architecture where authentication, authorization, and identity management are executed through Ethereum smart contracts. This decentralized approach improves device trustworthiness while minimizing opportunities for unauthorized access and identity spoofing, as reported by Jhariya and his colleagues and Vedashree and his colleagues [16, 17].

The developed system successfully integrated Ethereum, Solidity, Web3.js, React, and the Truffle development framework to provide a complete blockchain-based authentication solution. Smart contracts automated critical security operations including device registration, identity verification, digital signature validation, authorization, and device lifecycle management [8, 9, 13, 16].The automation of these processes reduced human intervention while ensuring that every authentication request was transparently recorded on the immutable blockchain ledger.

Experimental evaluation demonstrated that the BEDAAS system maintained stable authentication performance throughout repeated execution cycles. Authentication processing times remained relatively consistent across different sampling periods, indicating that the computational overhead introduced by blockchain operations is predictable and manageable for permissioned IoT environments. Although blockchain consensus contributed to higher processing time during transaction validation, the final authorization stage required considerably lower computational resources, demonstrating efficient separation between blockchain verification and access control functions.

The stability analysis further confirmed the robustness of BEDAAS. System convergence results showed that legitimate devices gradually reached stable trusted states, while suspicious or unauthorized devices were progressively rejected through continuous blockchain verification. This behavior indicates that the framework can effectively distinguish legitimate participants from malicious entities, thereby improving overall network resilience.

Communication latency analysis showed comparable performance between Geth and Infura Ethereum implementations, suggesting that the BEDAAS framework is platform-independent and can be deployed using either local blockchain nodes or cloud-based blockchain services. Furthermore, although authentication latency increased with increasing transaction volume, response times remained within acceptable operational limits for practical IoT applications. This demonstrates that the proposed framework scales efficiently while maintaining reliable authentication performance.

The integration of cryptographic security mechanisms, including SHA-256 hashing, Elliptic Curve Digital Signature Algorithm (ECDSA), Merkle Tree verification, Public Key Infrastructure (PKI), and blockchain consensus, significantly enhanced data integrity, authenticity, non-repudiation, and resistance against common cyberattacks such as replay attacks, impersonation attacks, message tampering, and unauthorized device access[2, 18].Every transmitted payload contained a digital signature and unique device identity that could be independently verified by the receiving IoT platform before data processing, thereby establishing a secure chain of trust throughout the communication process.

Hence, the findings demonstrate that BEDAAS provides a practical and secure blockchain-based authentication framework capable of improving confidentiality, integrity, availability, transparency, and accountability within IoT ecosystems.

5.Conclusion

This research successfully designed, implemented, and evaluated BEDAAS (Blockchain-Enabled Device Authentication and Authorization System) for securing Internet of Things networks. The proposed framework integrates blockchain technology with smart contracts to provide decentralized device authentication, authorization, and identity management without reliance on centralized authorities. The developed system demonstrated the feasibility of using Ethereum blockchain, Solidity smart contracts, Web3.js, React, and cryptographic security mechanisms to establish a secure trust framework for IoT devices. Experimental evaluation showed that the framework provides reliable authentication performance, efficient authorization, stable system behavior, and acceptable communication latency while maintaining strong protection against unauthorized access and identity-related attacks.

By combining blockchain immutability, digital signatures, cryptographic hashing, Merkle Tree verification, and smart contract automation, BEDAAS enhances the integrity, transparency, and traceability of IoT communications. The findings confirm that blockchain-enabled authentication represents a viable solution to many of the security challenges confronting conventional IoT infrastructures. Although computational overhead associated with blockchain consensus remains a limitation, the improved security, decentralization, and accountability offered by BEDAAS outweigh these performance costs, particularly in applications requiring high levels of trust and data integrity. Consequently, the proposed framework provides a strong foundation for future secure IoT architectures.

6.Limitations

1. Scalability constraints: The blockchain platforms used (e.g., Ethereum, Hyperledger) may not scale efficiently to billions of IoT devices due to transaction throughput limits and consensus overhead.
2. Resource consumption: IoT devices are resource-constrained (low power, limited memory). Running cryptographic operations and interacting with blockchain nodes can strain device capabilities.
3. Latency issues: Consensus mechanisms introduce delays. For real-time IoT applications (e.g., industrial control, healthcare monitoring), latency may hinder responsiveness.
4. Energy overhead: Blockchain transactions consume significant energy, which is problematic for battery-powered IoT devices and environmentally sensitive deployments.
5. Interoperability gaps: Integrating heterogeneous IoT devices and multiple blockchain platforms remains complex, limiting seamless adoption across diverse ecosystems.
6. Cost implications: Transaction fees, infrastructure costs, and maintenance overhead may limit deployment in resource-constrained environments.

7.Recommendations

The following recommendations are outlined for future research and further studies. These recommendations are:

1. Permissioned Blockchain Deployment: Future implementations should adopt permissioned blockchain platforms such as Hyperledger Fabric or Quorum for large-scale enterprise IoT environments where lower latency and higher throughput are required.

2. Lightweight Consensus Mechanisms: More efficient consensus algorithms such as Proof of Authority (PoA), Practical Byzantine Fault Tolerance (PBFT), or Delegated Proof of Stake (DPoS) should be investigated to reduce authentication delay and computational overhead.

3. Integration of Artificial Intelligence: Artificial intelligence and machine learning techniques should be incorporated into BEDAAS to enable intelligent anomaly detection, adaptive trust management, and predictive cyberattack identification.

4. Enhanced Privacy Protection: Advanced cryptographic techniques such as Zero-Knowledge Proofs (ZKPs), homomorphic encryption, and secure multi-party computation should be integrated to strengthen privacy preservation during authentication.

5. Comprehensive Scalability Evaluation: Future research should evaluate BEDAAS using thousands of heterogeneous IoT devices in real-world deployments to further assess scalability, fault tolerance, and resource utilization.

6 Industry Deployment: The proposed framework should be validated in practical environments, including smart healthcare, industrial IoT, intelligent transportation, smart agriculture, and smart city

infrastructures to assess long-term operational performance under real-world conditions.

References

[1] M. Stolpe, "The Internet of Things: Opportunities and Challenges for Distributed Data Analysis," *SIGKDD Explor. Newsl.*, vol. 18, no. 1, pp. 15–34, 2016, doi: 10.1145/2980765.2980768.

[2] M. Adam, M. Hammoudeh, R. Alrawashdeh, and B. Alsulaimy, "A Survey on Security, Privacy, Trust, and Architectural Challenges in IoT Systems," *IEEE Access*, vol. 12, pp. 57128–57149, 2024, doi: 10.1109/ACCESS.2024.3382709.

[3] S. S. Mahadik, P. M. Pawar, and R. Muthalagu, "Heterogeneous IoT (HetIoT) security: techniques, challenges and open issues," *Multimedia Tools and Applications*, vol. 83, no. 12, pp. 35371–35412, 2024/04/01 2024, doi: 10.1007/s11042-023-16715-w.

[4] R. Chataut, A. Phoummalayvane, and R. Akl, "Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0," *Sensors*, vol. 23, no. 16, p. 7194, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/16/7194.

[5] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications," *Automation in Construction*, vol. 101, pp. 111–126, 2019/05/01/ 2019, doi: https://doi.org/10.1016/j.autcon.2019.01.023.

[6] P. Bellini, P. Nesi, and G. Pantaleo, "IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies," *Applied Sciences*, vol. 12, no. 3, p. 1607, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1607.

[7] R. Tejushree, P., A. Prasad, R. Vedashree. C, V. R. Chandramule, K. Padimi, and N. Shwetha, "Secure E-voting: Leveraging Blockchain technology and Face recognition for enhanced authentication," in *2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N)*, 16–17 Dec. 2024 2024, pp. 1805–1810, doi: 10.1109/ICAC2N63387.2024.10895391.

[8] G. Iswarya and C. Vennila, "A secure and scalable blockchain-assisted authentication framework for decentralized IoT data management," *Scientific Reports*, 2026/04/04 2026, doi: 10.1038/s41598-026-47192-4.

[9] S. K. Panda, A. R. Sathya, and S. Das, "Bitcoin: Beginning of the Cryptocurrency Era," in *Recent Advances in Blockchain Technology: Real-World Applications*, S. K. Panda, V. Mishra, S. P. Dash, and A. K. Pani Eds. Cham: Springer International Publishing, 2023, pp. 25–58.

[10] P. Mueller, "Application of blockchain technology," *it - Information Technology*, vol. 60, no. 5-6, pp.

249–251, 2018, doi: doi:10.1515/itit-2018-0035.

[11] L. V. Vedashree, M. Sandhya, N. Shareeba, M. K. Sowmya, and R. Suravi, "Blockchain Based Authentication System for Internet of Things," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 13, no. 5, pp. 266–272, 2024, doi: 10.17148/IJARCCE.2024.13540

[12] S. Tahir, N. Tariq, F. A. Khan, H. Khan, and J. Al-Muhtadi, "A Secure Framework for Authenticating and Controlling Access to IoT Devices Integrated with Blockchain," in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, 11–14 Jan. 2025 2025, pp. 1–6, doi: 10.1109/ICCE63647.2025.10930226.

[13] A. K. Al Hwaitat *et al.*, "A New Blockchain-Based Authentication Framework for Secure IoT Networks," *Electronics*, vol. 12, no. 17, p. 3618, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/17/3618.

[14] W. Shujaa, M. Alanzi, and S. Sankaranarayanan, "Enhancing IoT security through blockchain integration," (in English), *Frontiers in Computer Science*, Review vol. Volume 7 - 2025, 2025–October–29 2025, doi: 10.3389/fcomp.2025.1670473.

[15] C. Walsh, P. O'Reilly, R. Gleasure, J. McAvoy, and K. O'Leary, "Understanding manager resistance to blockchain systems," *European Management Journal*, vol. 39, no. 3, pp. 353–365, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.emj.2020.10.001.

[16] M. K. Jhariya, V. Dehalwar, and J. Bharti, "SDIdA-IoT: self-sovereign digital identification and authentication framework for IoT devices using blockchain," *Cluster Computing*, vol. 28, no. 8, p. 495, 2025/08/19 2025, doi: 10.1007/s10586-025-05120-7.

[17] V. Vedashree, L., Y. Ramaswamy, L. Gudala, M. Abbas, H., and C. Sushama, "Privacy-Preserving Data Analysis using Decentralized Blockchain-Based Authentication with Zero-Knowledge Cloud Auditing," in *2025 3rd International Conference on Data Science and Information System (ICDSIS)*, 16–17 May 2025 2025, pp. 1–6, doi: 10.1109/ICDSIS65355.2025.11070587.

[18] S. R. Addula and A. Ali, "A Novel Permissioned Blockchain Approach for Scalable and Privacy-Preserving IoT Authentication," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 4, pp. 222–237, 2025, doi: 10.63180/jcsra.thestap.2025.4.3.