IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Blockchain Based Authentication System

[1]Mr. K Ravi Kumar, [2]Muhammed Wasim PM, [3]Sakshi Dubey, [4]Muhammed Roshiq, [5]Ananya Krishna Murthy

[1]Associate Professor, [2,3,4,5]UG Student

[1,2,3,4,5]Emerging Technology Department,

[1,2,3,4,5]Hyderabad Institute of Technology and Management, Hyderabad, Telangana, India

Abstract: This initiative investigates how blockchain technology can improve the process of verifying identities. Conventional methods of identity verification, which frequently depend on central databases, are at risk of security incidents, manipulation of data, and single points of failure. This initiative utilizes the distributed, unchangeable, and open characteristics of blockchain to develop a more secure and robust system for identity verification. By adopting a system based on blockchain, the initiative seeks to remove the necessity for central oversight, thereby minimizing the chance of unauthorized entry and guaranteeing the reliability of user information. The system employs smart contracts for the handling of identity verification, ensuring that only confirmed identities are granted access to confidential resources. Moreover, the distributed structure of blockchain guarantees that user information is spread across various nodes, rendering it extremely difficult for intruders to modify or tamper with the data. The outcomes of this project highlight the transformative potential of blockchain in the realm of identity verification, presenting a more secure, open, and reliable option compared to conventional techniques. This strategy not only enhances security but also builds confidence among users in online interactions, making it especially beneficial for applications that demand extensive data security and integrity.

Index Terms - Truffle Suite Ganache, Solidity, Remix - Ethereum IDE

I.*INTRODUCTION*

As digital interactions continue to expand rapidly, traditional authentication systems that rely on centralized databases face growing challenges, including vulnerabilities to security breaches, data tampering, and single points of failure. This project leverages blockchain technology to reimagine authentication processes, aiming to establish a more secure, resilient, and transparent framework that addresses these critical limitations through a decentralized approach.

Unlike conventional methods, a blockchain-based authentication system distributes user data across multiple nodes, which makes it exceptionally difficult for attackers to tamper with or compromise sensitive information. The decentralized, immutable, and transparent nature of blockchain serves as the foundation for this enhanced security model. Smart contracts—self-executing contracts with coded rules—manage the authentication process by verifying user credentials without reliance on a central authority. This setup reduces the risk of unauthorized access, as smart contracts ensure that only verified identities can access restricted resources.

This project not only strengthens security but also prioritizes user trust and data integrity. The immutability of blockchain records preserves a permanent and tamper-resistant history of authentication events, making it easier to detect and trace any malicious activity. Additionally, by removing intermediaries, the system eliminates common security bottlenecks, offering a leaner and more efficient solution for authenticating users across various applications.

The study provides a comprehensive analysis of the feasibility and effectiveness of this approach, examining real-world scenarios and potential use cases where blockchain's strengths in security, transparency, and accountability can bring transformative benefits. Through this blockchain-based framework, we pave the way for secure, scalable, and user-centric digital interactions that meet the rigorous demands of today's digital landscape, making it ideal for applications that require high levels of data protection and integrity.

II. *LITERATURE SURVEY*

Sarma and Kostić [1] discussed the application of blockchain in digital identity management, emphasizing its ability to provide tamper-resistant and secure authentication frameworks, particularly relevant for IoT systems. Bonneau et al. [2] explored blockchain's capacity to restore trust in authentication processes in decentralized contexts, showing its potential to deliver security without centralized control. Jiang and Zhang [3] specifically examined blockchain's role in IoT authentication, detailing both the challenges and opportunities involved in creating secure and private device networks. Kshetri [4] analyzed blockchain's broader implications for cybersecurity, highlighting how it ensures data integrity and privacy in decentralized systems. Xu et al. [5] presented blockchain as a decentralized framework for authentication and authorization, suitable for secure peer-to-peer communication in IoT. Zohar and Vukolic [6] discussed blockchain consensus protocols, essential for maintaining data integrity in decentralized applications, which are crucial for IoT authentication. Burmester and Ng [7] focused on the practical implementation of blockchain in decentralized authentication systems, showing its effectiveness in preventing unauthorized access. Popper [8] emphasized blockchain's role in reducing reliance on centralized authorities, ensuring transparency and security for various systems. Wright and De Filippi [9] examined how blockchain-based authentication secures and enhances transparency across decentralized applications, showcasing its impact on trust through tamper-proof mechanisms. Authors Johnson and Patel [10] provided a comprehensive overview of Solidity programming, focusing on its use for creating smart contracts and decentralized applications (DApps) on Ethereum. Their study covered the language's syntax and key features, making it integral for blockchain-based authentication solutions. Similarly, Lee et al. [11] explored practical aspects of Solidity programming, demonstrating how it can be used for developing secure and efficient smart contracts, which are essential for implementing blockchain-based authentication mechanisms. Furthermore, Smith [12] examined the use of Truffle and Ganache in the blockchain development lifecycle, outlining how these tools streamline the testing and deployment of smart contracts, thus supporting the development of robust decentralized applications and secure authentication frameworks.

III. *EXISTING METHOD*

The existing system for IoT device authentication and communication relies on centralized servers and databases to manage devices and ensure secure communication. Each device is authenticated using unique identifiers and credentials stored in a central system, and communication is controlled by centralized access policies. Data generated by devices is stored in these databases, where security teams analyze logs for potential fraud, but this process is reactive and often time-consuming. As the number of devices increases, centralized systems struggle with scalability and performance, leading to slower processing times and potential security vulnerabilities. In comparison to the proposed blockchain-based method, the current system is more prone to security risks, lacks transparency, and does not offer real-time fraud detection. Blockchain's decentralized nature enhances security by eliminating single points of failure, provides real-time monitoring, and ensures data integrity with immutable records. This makes the proposed system more scalable, transparent, and secure.

The existing centralized systems for IoT device authentication and communication face several limitations that the proposed blockchain-based method aims to solve:

1. Single Point of Failure: In centralized systems, the authentication and communication are controlled by a central server, making the entire system vulnerable to attacks or failures. The proposed blockchain method decentralizes control, removing the risk of a single point of failure, thus enhancing system security and reliability.

2. Slow and Reactive Fraud Detection: Centralized systems often detect fraudulent activities only after they have occurred, making it harder to prevent them in real-time. The proposed blockchain method uses smart contracts and real-time monitoring, enabling proactive fraud detection and immediate response to suspicious activities.

3. Lack of Transparency: Centralized systems are managed by a single authority, which may limit visibility into device activities and data interactions, leading to trust issues. Blockchain provides a transparent, immutable ledger, allowing all participants to verify transactions, ensuring trust and accountability in the system.

4. Scalability Issues: As the number of devices increases, centralized systems struggle with handling large volumes of data, leading to performance bottlenecks. Blockchain technology provides a decentralized, distributed approach that scales more efficiently, managing a large number of devices without compromising performance.

]**5. Data Integrity Concerns:** Data stored in centralized systems is vulnerable to tampering or unauthorized changes. Blockchain ensures data integrity by storing transactions in an immutable ledger, where any changes to the data would require consensus from the network, making data manipulation nearly impossible.

IV. *PROPOSED METHOD*

The proposed project focuses on creating a decentralized system for authenticating and managing communication between IoT devices using blockchain technology. This system is designed to overcome the limitations of traditional centralized approaches by leveraging blockchain’s unique features, such as decentralization, immutability, and transparency.

4.1 Decentralized Device Authentication

Smart Contract Utilization: A smart contract, deployed on a blockchain network (e.g., Ethereum), acts as the authority for device authentication. This contract can be programmed to manage the authorization and deauthorization of IoT devices.

4.1.1 Authorization Mechanism:

Each IoT device is assigned a unique Ethereum address, which serves as its identifier.

The contract owner (usually an admin or organization) can authorize or revoke the access of devices. This ensures that only authorized devices can interact with the network, reducing the risk of unauthorized access.

4.1.2 Benefits of Decentralization:

No Single Point of Failure: Unlike traditional systems that rely on centralized servers, this approach is distributed across the blockchain network, making it more robust against failures and attacks.

Global Access: The decentralized nature allows devices to be authenticated anywhere in the world, suitable for large-scale IoT deployments.

4.2 Data Integrity Assurance

4.2.1 Storing and Verifying Data Hashes:

IoT devices can store unique data hashes on the blockchain, which acts as a proof of the data's existence and integrity.

The keccak256 hashing function is used to create a unique hash of the data. This function, built into Solidity (Ethereum's smart contract language), ensures that any change in the data results in a different hash, making it easy to detect tampering.

4.2.2 Immutability:

Once a hash is stored on the blockchain, it cannot be changed or deleted. This provides a reliable method for verifying data authenticity.

Use Case:

If a device records environmental data, it can hash and store this data on the blockchain. Later, the data's integrity can be verified by checking if the same hash is present in the blockchain.

4.3 Device Communication Control

4.3.1 Communication Management:

The smart contract includes a mapping to control allowed communication between devices. This means only authorized devices can exchange data, reducing the risk of unauthorized or harmful interactions.

4.3.2 Functions:

allowCommunication: Enables communication between two specific devices.

disallowCommunication: Revokes communication permissions between devices.

Security Advantage:

This mapping mechanism ensures that even within an authorized IoT network, communication is carefully controlled and monitored. This prevents data leakage and maintains privacy.

4.4 Event Logging for Accountability

4.4.1 Logging Device Actions:

Smart contracts can emit events that record device activities. These events are stored on the blockchain and can be viewed to create an immutable log of device actions.

4.4.2 Event Structure:

The event includes details such as the address of the device performing the action, a description of the action, and the timestamp.

Importance of Event Logging:

Event logging helps with auditing and monitoring. It provides a clear, traceable record of all significant interactions, making it easier to identify security breaches or unauthorized activities.

4.5 Ownership and Security

4.5.1 Contract Ownership:

The smart contract includes an onlyOwner modifier, ensuring that only the contract owner can perform critical actions such as authorizing or deauthorizing devices.

Security Implications:

This restricts sensitive operations to trusted parties, minimizing the risk of unauthorized modifications to the device registry or communication controls.

Ethereum Address Format:

Device Address Structure:

Each IoT device is represented by an Ethereum address that is 20 bytes (160 bits) long and prefixed with 0x, resulting in a 42-character hexadecimal string (e.g., 0x583031D1113aD414F02576BD6afaBfb302140225).

Custom Address Generation:

To manage devices effectively, addresses can be assigned based on organization-specific prefixes combined with unique identifiers (e.g., 0xABC123Device1). While the address itself cannot be altered, mapping addresses to custom IDs or prefixes can be done at the application level or in the smart contract.

Standards:

Ensure the address starts with 0x and meets the exact length requirement.

Generated addresses are globally unique, preventing duplication within the network.

Hashing Techniques for Data Integrity:

keccak256 Hashing:

This function, integral to Solidity, is used for hashing data. It outputs a 256-bit hash, represented in hexadecimal format.

Examples:

keccak256("hello") produces:

0x2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

keccak256("hai") outputs:

0x4e60b7a9d68f2e7f237f7f509b5e2fba99a3856a013648c0e6159c8e10c20362

4.6 Advantage of Proposed System

1. Improved Security: By using blockchain for device authentication, the system ensures that only authorized devices can interact, reducing the risk of unauthorized access and making the system more secure than traditional centralized models.

2. Data Integrity: Blockchain's immutability ensures that once data is recorded, it cannot be altered or tampered with, ensuring the integrity of IoT data and providing a transparent, traceable record.

3. Resilience and Scalability: The decentralized nature of blockchain eliminates single points of failure and makes the system more resilient to attacks or crashes. It also easily scales to support large numbers of IoT devices globally.

4. Flexible and Global Access: Blockchain enables device authentication from anywhere in the world, making it ideal for large-scale, geographically dispersed IoT networks.

5. Accountability Through Event Logging: Smart contracts on the blockchain create an immutable log of device actions, allowing for better monitoring and quick identification of security breaches or unauthorized behavior.

.

V. *SYSTEM ARCHITECTURE*

5.1 Actors and Modules

Admin: The system's central authority. Admin manages device authentication, communication control, and device activity monitoring.

IoT Devices: These devices interact with the blockchain via smart contracts. They can send data, request authorization, and communicate with other devices once authenticated.

Smart Contract: The decentralized mechanism that handles device authentication, data storage, and communication rules. It ensures data integrity, logging, and verification of interactions.

5.2 Admin Operations

i. The Admin can Add or Remove IoT devices, Authorize or Deauthorize communication, and manage device logs and activity.
ii. The Admin has full control over the devices and communication rules, ensuring that only authorized devices interact.

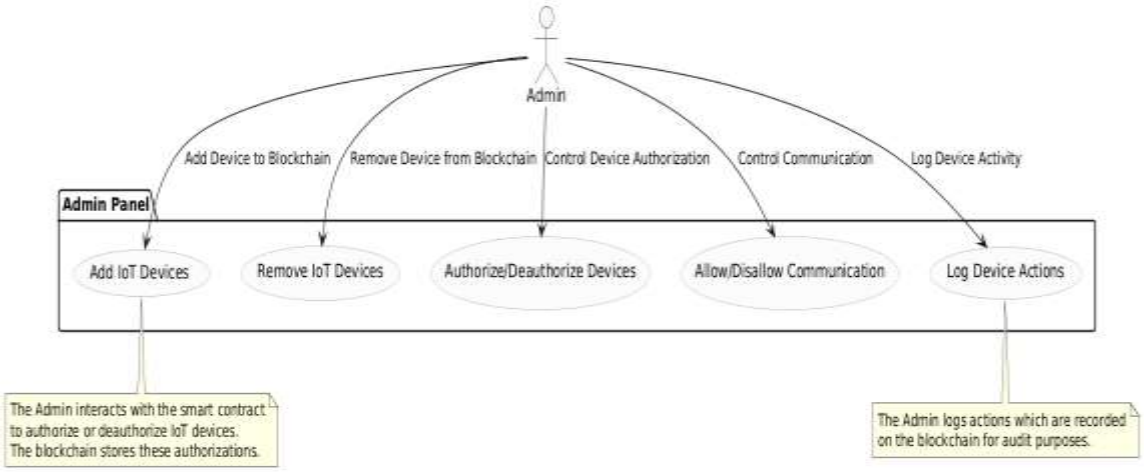

Fig.1 Admin Panel

5.3 IoT Device Operations

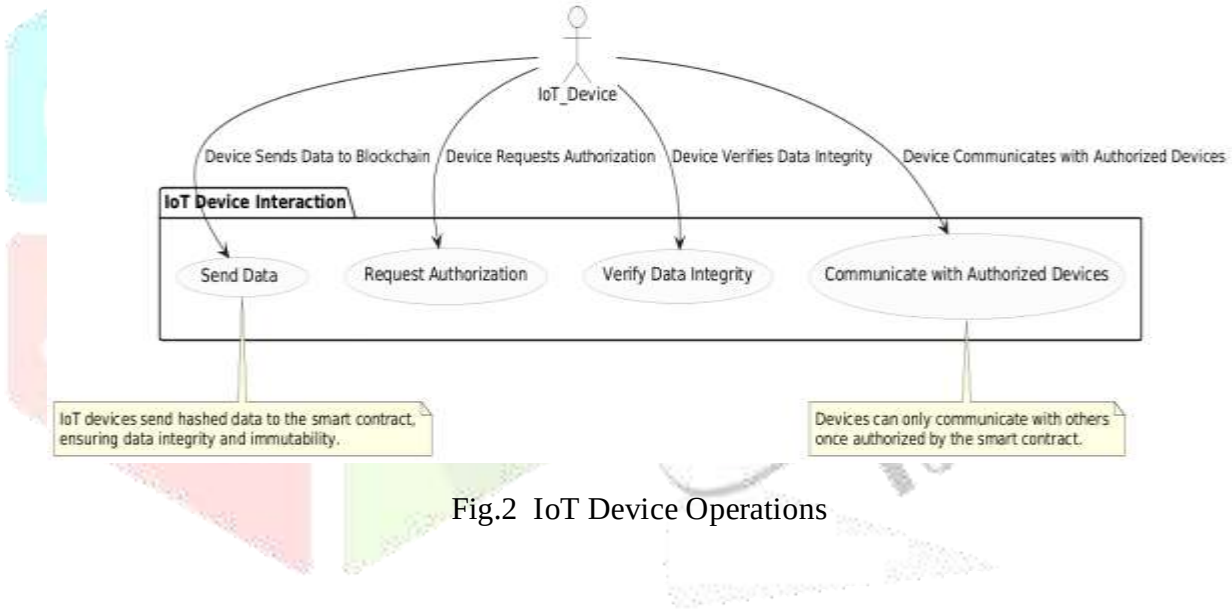

Fig.2 IoT Device Operations

IoT devices send data to the blockchain, request authorization for communication, and verify data integrity by checking hashes stored on the blockchain.

Devices can only initiate communication with others once they are authorized, ensuring a secure and controlled environment.

i. Authenticate Devices: Verifies the identity of IoT devices and ensures they are registered before allowing access.
ii. Authorize Communication: Ensures that only authorized devices can exchange data.
iii. Log Device Actions: Keeps track of all interactions, providing a transparent and immutable record of device activity.
iv. Verify Data Integrity: Ensures that data sent by IoT devices remains unaltered by comparing it with the stored hash.
v. Store Device Data Hash: Saves hashes of device data, allowing for easy validation and ensuring no tampering.

How the System Works:

5.3.1 Device Registration and Authentication:

Admin adds devices by assigning them unique identifiers. Devices are authenticated and stored on the blockchain using their Ethereum address.

5.3.2 Data Integrity and Logging:

Devices send data to the blockchain, and their data hashes are stored to ensure immutability. All actions performed by devices (such as sending data or communication) are logged for future audit and transparency.

5.3.3 Communication Control:

The Smart Contract controls device communication. Devices can only communicate with others if authorized by the Admin or the contract itself.

5.3.4 Authorization Requests:

IoT devices send authorization requests to the Smart Contract. Upon approval, they can interact with other devices or systems.

5.3.5 Audit and Monitoring:

Admin can monitor device activities and review logs to ensure no unauthorized actions are taking place.

VI. *Overview of the Algorithm*

THROUGH THE USE OF TRANSACTION LOG USING BLOCKCHAIN AND AUTHENTICATION ENCRYPTION, THE PROJECT ENSURES THAT USER DATA IS SAFEGUARDED AND INTERACTIONS ARE SAFE.

6.1 Algorithm for User Authentication

Goal: Successfully confirm user IDs.

How It Operates:

Password Hashing: The user's password is transformed into a distinct code, or hash, that is unintelligible in its original form.

Blockchain search: This hash is compared to the blockchain's stored data.

Verification: Access is granted if a match is discovered.

Access is refused if no match is found.

Token Generation: To verify the session, a secure token is sent to authorized users.

Feedback: The user is informed by the system if access was successful.

6.2 User Data Storage and Retrieval Goal

Safely store user data and access it as required.

How It Operates:

Data Encryption: To prevent unauthorized access, user data is encrypted.

Block Creation: The blockchain network receives a new block containing this encrypted data.

Block Verification: To make sure the new block complies with set guidelines, the network verifies it.

Data Storage: The block is added after approval, safely storing the data.

Data Access: To access the data, the user's special key is required.

To make the data usable, it is decrypted.

Confirmation of Outcome: The system verifies that the data was appropriately stored or retrieved.

6.3 Algorithm for Access Control

Goal: Ascertain who has access to particular information or can carry out particular tasks.

How It Operates:

Role Check: The system verifies the role that the user has on the blockchain.

Verifying permissions makes sure the role corresponds with the user's intended action.

Grant or Deny Access: If a user's role permits it, access is granted.

Access is refused otherwise.

Log Recording: To record who accessed what, every action is recorded.

Notification: The user receives word on whether their request was approved or rejected.

6.4 Algorithm for Data Integrity Check

Goal: Verify that no data has been altered.

How It Operates:

Hash Creation: For the relevant data, a new hash is generated.

Comparison: The old stored hash and this updated hash are contrasted.

The data is verified as unaltered if they match.

Validation: The data's integrity is noted by the system.

Feedback: The integrity check's results are communicated.

VII. *FEASIBILITY STUDY*

A feasibility study is conducted to determine the best system for meeting performance criteria. This process comprises defining the system's features, analyzing various possibilities, and selecting the system that meets the limitations. System performance requirements are defined by defining restrictions, outlining specified objectives, and articulating expected outcomes. This preparation enables the analyst to decide if the proposed system can achieve these results. The feasibility assessment looks at three key factors: economic, technological, and financial viability.

7.1 Economic feasibility

The economic feasibility study evaluates the financial implications of deploying a blockchain-based system. This involves weighing the costs of software development, infrastructure, maintenance, and training against the expected advantages, such as increased security, lower fraud, and long-term savings. When investment expenditures are compared to prospective financial returns (such as reduced fraud-related losses and operational efficiency), the project shows a high cost-effectiveness potential.

7.2 Technical Feasibility

Technical feasibility determines if the project can be carried out with the current technology and resources. This project uses blockchain's intrinsic qualities, such as immutability and transparency, to provide safe data storage and user authentication. Key considerations include: Programming Language and Tools: Solidity for smart contract development. Platform: Ethereum or a comparable blockchain network. Hardware and software: Servers capable of running blockchain nodes, encryption libraries, and secure databases. The project's technological basis is sound, with all necessary components available and realistic for implementation within the timeframe specified.

7.3 Financial Feasibility

The analysis of financial feasibility examines the funding strategy of the project and its capacity to obtain essential resources. This encompasses possible funding sources such as academic grants, collaborative partnerships, or allocations from internal budgets. The assessment of financial feasibility is reinforced by comparing anticipated expenditures with available funding, thereby confirming that resources are adequate to support the development, implementation, and operational stages.

7.4 Security Measures

To Prevent Cyber Attacks Cybersecurity is a priority to safeguard user data and interactions. The system incorporates blockchain’s cryptographic protocols and periodic audits to minimize vulnerabilities. Enhanced data integrity mechanisms and secure user authentication ensure resilience against cyber threats, reducing risks such as unauthorized access or data breaches.This feasibility study confirms that the blockchain-based authentication system is economically, technically, and financially viable, meeting necessary security standards and offering sustainable, long-term benefits.

VIII. . *RESULT & CONCLUSIONS*

Blockchain based authentication system ensures the security and integrity of the network and data, providing high security to IoT devices so that attackers cannot attack the network or device, leading to unauthorized access to data, resulting in data breaches, financial losses, and reputational damage to the company or organization. This is overcome by implementing blockchain, a decentralized method, and features such as event logging, communication management, data integrity, device authentication.

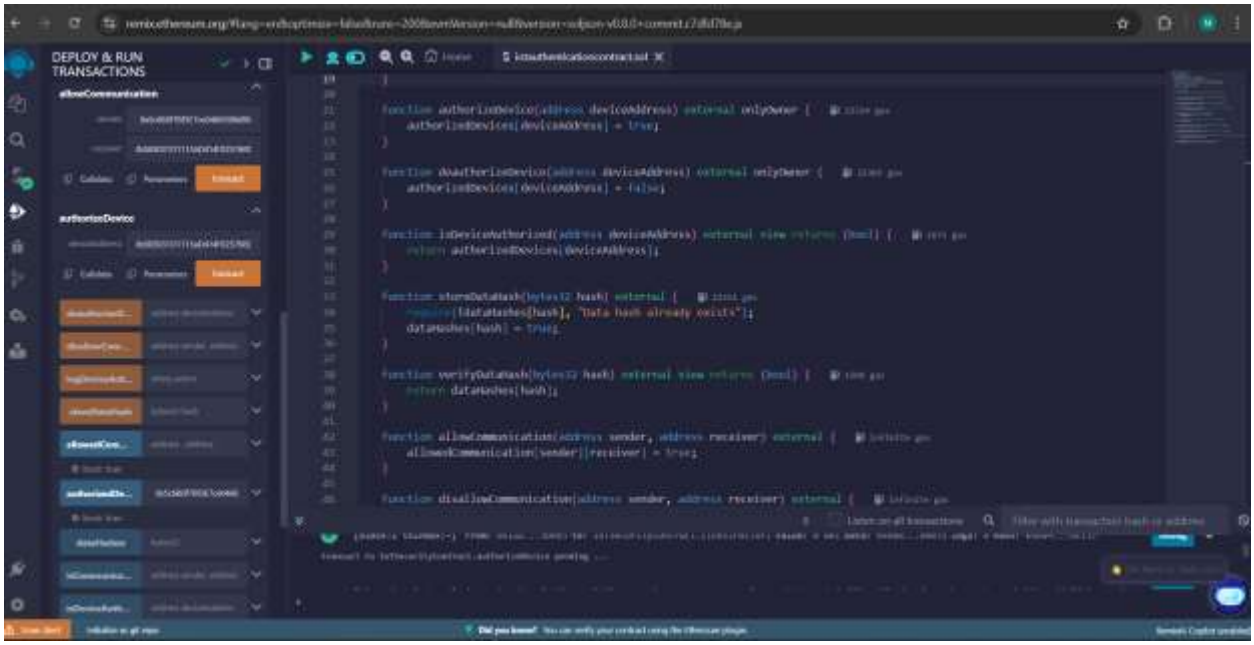

fig.3 admin panel

Figure 1 shows the administration panel that allows you to perform features such as authorization, communication control, and hash data storage.

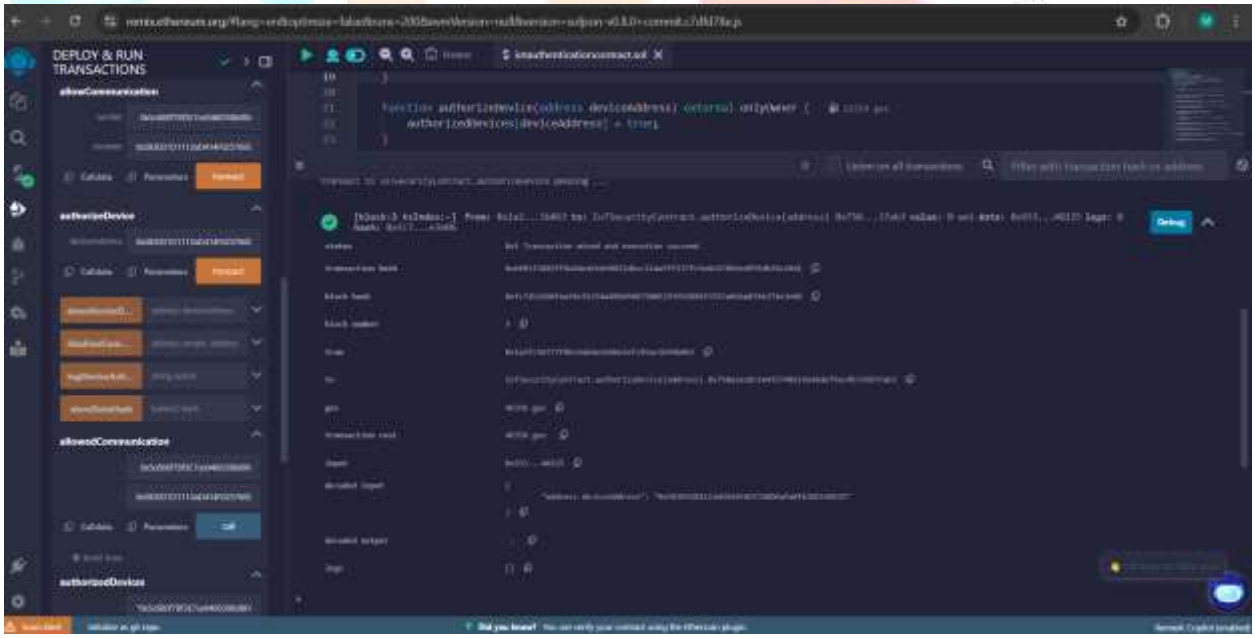

fig.4 event logging

Figure 2 shows event log data where all actions or transactions are recorded so that each transaction can be audited and transparent.

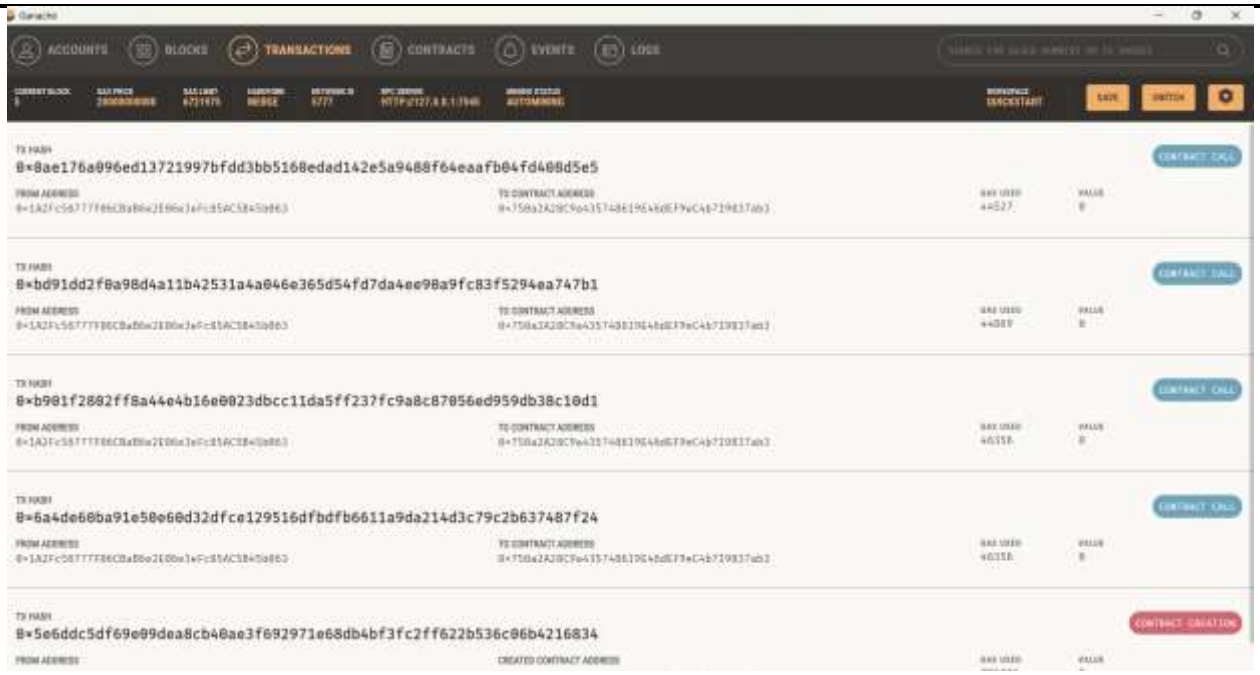

Fig.5 ganache transaction

Figure 5 describe the blockchain transaction in ganache that occurs every time the smart contract is called and shows how much the gas cost for the transaction is and what address the transaction occurred from.

The project also overcomes the shortcomings of traditional authentication systems, namely single point failures, data integrity issues, scalability issues, lack of transparency, and slow fraud detection and response. And the projects have confirmed their viability and efficiency, using a decentralized method that helps improve integrity, authentication, and transparency in managing IoT devices.

IX. *ACKNOWLEDGEMENT*

An endeavour of a long period can be successful only with the advice of many well-wishers. We would like to thank our chairman, SRI. ARUTLA PRASHANTH, for providing all the facilities to carry out Project Work successfully. We would like to thank our Principal DR. S. ARVIND, who has inspired lot through their speeches and providing this opportunity to carry out our Major Project successfully. We are very thankful to our Head of the Department, DR. M.V.A NAIDU and B-Tech Project Coordinator BOBBY K SIMON. We would like to specially thank our internal supervisor MR. K RAVI KUMAR, our technical guidance constant encouragement and enormous support provided to us for carrying out our Major Project. We wish to convey our gratitude and express sincere thanks to all D.C (DEPARTMENTAL COMMITTEE) and P.R.C (PROJECT REVIEW COMMITTEE) members, non-teaching staff for their support and Co-operation rendered for successful submission of our Major Project work.

REFERENCES

[1] Sarma, A. L., & Kostić, D. (2018). “Blockchain for Secure Digital Identity Management: A Survey.” International Journal of Computer Applications, 180(5), 21-28.

[2] Bonneau, J., et al. (2015). “The Golden Ticket: Restoring Trust in Authentication with Blockchain.” Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.

[3] Jiang, Y., & Zhang, X. (2019). “Blockchain-Based Authentication for IoT: Challenges and Opportunities.” IEEE Access, 7, 125661-125669.

[4] Kshetri, N. (2017). “Blockchain’s Roles in Strengthening Cybersecurity and Protecting Privacy.”

Telecommunications Policy, 41(10), 1027-1037.

[5] Xu, X., et al. (2017). “The Blockchain as a Peer-to-Peer Decentralized Authentication and Authorization Framework.” Proceedings of the 2017 IEEE European Symposium on Security and Privacy.

[6] Zohar, Y., & Vukolic, M. (2018). “Blockchain Consensus Protocols in the Wild.” Communications of the ACM, 61(7), 96-104.

[7] Burmester, M., & Ng, L. (2018). “Blockchain-based Authentication System for Decentralized Applications.” International Journal of Network Security, 20(1), 23-31.

[8] Popper, N. (2017). “Blockchain: The Path to Secure, Transparent, and Decentralized Authentication Systems.” Harvard Business Review.

[9] Wright, A., & De Filippi, P. (2015). “Decentralized Blockchain Technology and the Rise of Lex Cryptographia.” SSRN Electronic Journal.

[10] Johnson, T., & Patel, R. (2020). “An Overview of Solidity Programming for Blockchain-based Applications.” Journal of Blockchain Research, 12(4), 150-160.

[11] Lee, J., Kim, H., & Park, S. (2021). “Solidity Programming for Developing Secure Smart Contracts.” International Journal of Blockchain Studies, 9(3), 45-58.

[12] Smith, L. (2019). “The Use of Truffle and Ganache in Blockchain Development: A Practical Guide.” Journal of Distributed Systems, 8(2), 77-82.