Smart Product Authentication Using Blockchain with Cloud Integration

A. Yugandhara Rao1, P. Mahathi Roy2, S. Santhosh Raju3, S. Jayasree4, P. Uday Kumar5
[1,2,3,4]*Computer Science and Engineering, Lendi Institute of Engineering and Technology*
[5]*Associate Professor, Computer Science and Engineering, Lendi Institute of Engineering and Technology*

***Abstract*—Counterfeit products have become a major problem affecting customers and industries. The existing systems use static QR codes for product verification, which can be easily copied or misused. To overcome this, our project proposes a Smart Product Authentication System using Blockchain that ensures secure and tamper-proof verification of products. In this system, every product is registered on the blockchain network with a dynamic QR code and an NFC tag. The dynamic QR changes with each transaction, preventing duplication, while the NFC tag provides additional security through chip-based authentication. When the customer scans the QR or taps the NFC tag, the system verifies the product details stored on the blockchain, ensuring transparency and authenticity. This method builds trust between manufacturers and consumers by providing a secure and decentralized way to identify genuine products and eliminate fake ones.**

***Index Terms*—Blockchain, Dynamic QR Code, NFC Tag, Product Authentication, Smart Contract.**

I. INTRODUCTION

Counterfeit products have become a serious problem in today's market, affecting both consumers and manufacturers. Fake products not only reduce brand value but also create safety risks for users. Traditional product authentication systems mainly depend on QR codes or centralized databases, which can be easily duplicated or manipulated, making them unreliable for secure verification. With the advancement of technology, there is a need for a more secure and transparent system that ensures product authenticity. Blockchain technology provides a decentralized and tamper-proof platform where product data can be securely stored and verified. It ensures that once the data is recorded, it cannot be altered, thereby increasing trust among stakeholders. In this project, a Smart Product Authentication System using Blockchain with Cloud Integration is proposed. The system uses QR codes and NFC tags for unique product identification and allows users to verify product authenticity in real time. Cloud storage is used to manage large data efficiently, while blockchain ensures data security and transparency. This approach helps in reducing counterfeit products and improving trust in the supply chain.

II. SYSTEM ARCHITECTURE

A System Architecture defines the overall structure and organization of a system, detailing its components, their interactions, and how they work together to achieve the system's goals. A System Architecture defines the overall structure and organization of a system, detailing its components, their interactions, and how they work together to achieve the system's goals. In the Smart Product Authentication System using Blockchain with Cloud Integration, the architecture includes the User Layer (Manufacturer, Seller, Consumer, Regulator), Application Layer (React.js frontend), Backend Module (Node.js/Express API server), Blockchain Layer (Ethereum smart contracts), Cloud Storage (IPFS), Identification Modules (QR code and NFC), Database (PostgreSQL), and Verification Layer (hash comparison and authentication result). It outlines the flow from product registration to final verification, ensuring security, scalability, and transparency in product authentication.

The system architecture is designed to provide secure and efficient verification of products using blockchain and cloud technologies. The process begins with the User Layer, where different users interact with the system through a web-based interface. Manufacturers register products and upload details, sellers manage product distribution and ownership transfer, consumers verify products, and regulators monitor transactions. The data

entered by users is transmitted to the backend server through the frontend application.

The Application Layer consists of a React.js-based web interface that provides role-based dashboards for different users. It allows product registration, QR scanning, NFC interaction, and result visualization. This layer ensures smooth communication between users and the backend through REST APIs, providing a user-friendly and responsive experience.

At the core of the system is the Backend Module, implemented using Node.js and Express.js. It handles API requests, user authentication using JWT, QR code generation and decoding, NFC UID processing, and SHA-256 hashing of product data. It also interacts with the PostgreSQL database to store user information, product records, and transaction logs, ensuring efficient data management.

The Blockchain Layer uses Ethereum and Solidity smart contracts to store immutable product records and transaction details. Each product data stored in IPFS is hashed, and the hash is recorded on the blockchain, ensuring tamper-proof storage. MetaMask is used for wallet authentication and secure execution of blockchain transactions such as product registration and ownership transfer.

The Cloud Storage Layer is implemented using IPFS, where large product-related data such as certificates, images, and details are stored. Instead of storing complete data on the blockchain, only the hash of the IPFS data is maintained on the chain. During verification, the system retrieves the data from IPFS and compares its hash with the blockchain record to ensure data integrity.

The system concludes with the Verification and Output Layer, where the product's authenticity is determined. When a consumer scans the QR code or taps the NFC tag, the system retrieves the product data, computes its hash, and compares it with the blockchain hash. If the values match and the NFC UID is valid, the product is declared authentic; otherwise, it is marked as fake. The result is displayed on the user interface, and all verification activities are recorded on the blockchain, enabling a transparent audit trail for regulators.

The Smart Product Authentication System architecture is designed to ensure secure, scalable, and reliable product verification. It comprises the following components:

- User Layer:

Includes Manufacturer, Seller, Consumer, and Regulator, each performing specific roles such as product registration, ownership transfer, verification, and monitoring.

- Application Layer (React.js):

Provides a web-based interface with role-based dashboards for user interaction and system access.

- Backend Module (Node.js/Express):

Handles API communication, authentication, QR/NFC processing, and SHA-256 hashing operations.

- Blockchain Layer (Ethereum):

Stores immutable product records and transaction history using Solidity smart contracts.

- Cloud Storage (IPFS):

Stores large product data securely and provides hash-based referencing for blockchain integration.

- Database (PostgreSQL):

Manages user data, product details, and transaction logs efficiently.

- Identification Modules:

Includes QR code generation and NFC UID-based product identification for secure verification.

- Verification Module:

Compares blockchain hash with IPFS data and validates NFC UID to determine product authenticity.

This architecture ensures a streamlined, secure, and transparent system for product authentication, effectively reducing counterfeit products and improving trust in the supply chain.

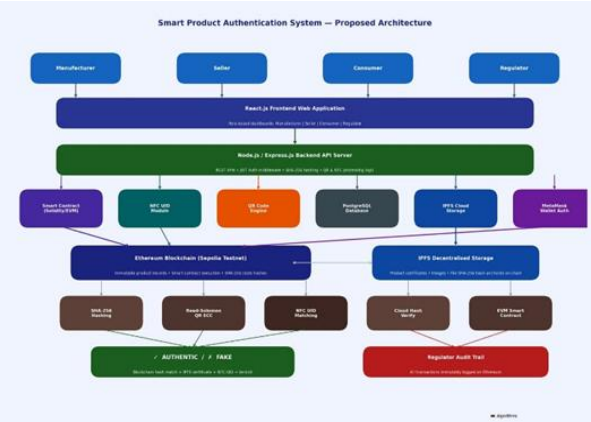

Fig- 1: System Architecture

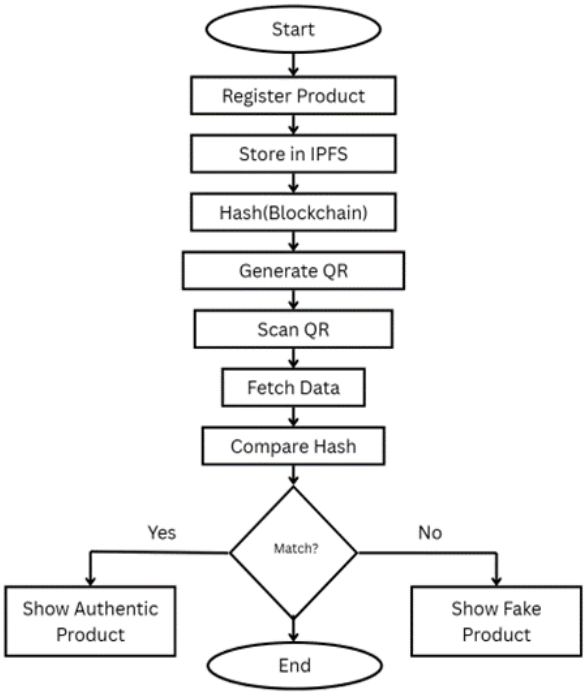

Fig -2: Flow Chart

III. TECHNOLOGIES AND TOOLS USED

- JavaScript:

JavaScript is used as the primary programming language to develop both frontend and backend components, enabling dynamic and interactive web functionality.

- Node.js:

Node.js is used for backend development to handle server-side logic, API requests, and communication between the frontend, blockchain, and database.

- React.js:

React.js is used to build a user-friendly frontend interface for manufacturers, sellers, and consumers to interact with the system efficiently.

- Solidity:

Solidity is used to develop smart contracts on the Ethereum blockchain, which store and manage product data in a secure and tamper-proof manner.

- Ethereum Blockchain:

Ethereum provides a decentralized platform for executing smart contracts and maintaining immutable product records.

- MetaMask:

MetaMask is used as a digital wallet to connect users to the blockchain and perform secure transactions such as product registration and ownership transfer.

- IPFS (Interplanetary Filesystem):

IPFS is used for decentralized cloud storage to store large product-related data, while only the hash is stored on the blockchain for integrity verification.

- QR Code Technology:

QR codes are generated for each product to enable quick and easy product verification by consumers.

- NFC Technology:

NFC tags are used to assign a unique identity to each product, allowing secure and reliable product tracking.

- PostgreSQL/Database:

A database is used to manage user data, transaction records, and system-related information efficiently.

IV. MODULES AND ALGORITHMS

- Manufacturer Module:

This module allows manufacturers to register on the system and add product details. The product information is stored in cloud storage, and its hash is recorded on the blockchain to ensure data integrity and security.

- Seller Module:

The seller module enables verified sellers to access product details, manage inventory, and transfer product ownership securely using blockchain transactions.

- Consumer Module:

Consumers can verify product authenticity by scanning the QR code or using an NFC tag. The system checks the product data with blockchain records and displays whether the product is genuine or fake.

- Blockchain Module:

This module manages smart contracts and stores immutable product records on the blockchain. It ensures transparency, security, and tamper-proof data storage.

- SHA-256 Algorithm:

Used to generate secure hash values for product data stored in the cloud. These hash values are stored on the blockchain to verify data integrity.

- QR Code Encoding and Decoding Algorithm:

Used to generate QR codes for each product and decode them during verification. It enables quick access to product information linked with blockchain records.

- Smart Contract Execution (Ethereum):

Smart contracts are executed on the Ethereum blockchain to handle product registration, ownership transfer, and verification in a secure and automated manner.

- Cloud Hash Verification Algorithm:

This algorithm compares the hash of the cloud-stored data with the hash stored on the blockchain to ensure that the product data has not been tampered with.

These modules and algorithms collectively enable efficient and accurate counterfeit product detection.

V. IMPLEMENTATION

The implementation of the Smart Product Authentication System using Blockchain with Cloud Integration begins with setting up the development environment using Node.js for backend services, React.js for frontend development, and Ethereum tools such as MetaMask and Truffle for blockchain interaction. The system is designed as a web-based application where users interact through a graphical interface, while backend APIs manage communication between the blockchain, cloud storage, and database. A database is configured to store user information, product records, and transaction details securely. Initial system setup also includes configuring IPFS for decentralized cloud storage and integrating it with the backend for efficient data handling.

The product registration process is initiated by the manufacturer, who connects to the system using a blockchain wallet such as MetaMask. After authentication, the manufacturer enters product details and uploads supporting documents or certificates. These details are stored in IPFS, generating a unique content hash. This hash is then recorded on the Ethereum blockchain using smart contracts written in Solidity, ensuring immutability and tamper-proof storage. Each product is assigned a unique identity, and a QR code is generated that links to the stored data. Additionally, an NFC tag is associated with the product to provide an extra layer of secure identification.

The system also supports seller operations, where registered sellers can access product details and perform ownership transfers. When a product is transferred from the manufacturer to the seller or from the seller to the consumer, the transaction is executed through smart contracts and recorded on the blockchain. This creates a transparent and traceable record of the product lifecycle. The backend ensures that only verified users can perform such operations, maintain system security and prevent unauthorized access.

For product verification, the consumer scans the QR code or taps the NFC tag using a mobile device. The system retrieves the corresponding product data from IPFS and computes its hash. This hash is then compared with the hash stored on the blockchain. If both values match, the system confirms that the product is genuine; otherwise, it flags the product as counterfeit. The result is displayed.

The entire system is deployed on a local server environment and can be extended to cloud platforms for better scalability and accessibility. Performance optimizations such as efficient API handling and lightweight smart contract execution ensure quick response times during verification. The implementation successfully integrates blockchain, cloud storage, and identification technologies to provide a secure, transparent, and scalable solution for product authentication, effectively reducing the risks associated with counterfeit products.

VI. RESULTS

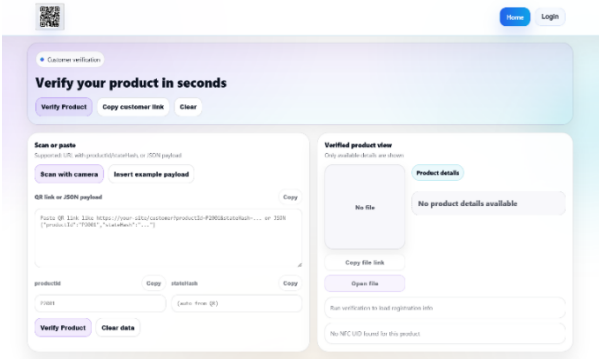

Fig -3: *Consumer Interface*

This is the interface where the Customer can access this for product results.

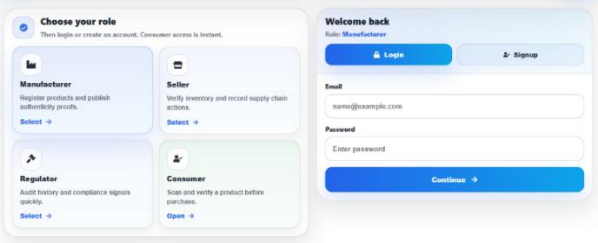

Fig -4: *Signup/Login Interface*

Login and role selection interface for secure user access in the product authentication system.

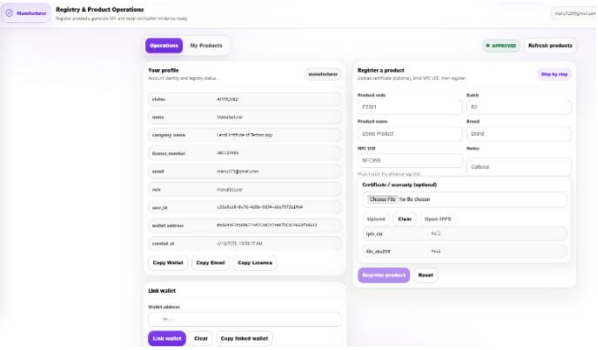

Fig -5: *Wallet Connection & Product Registration*.

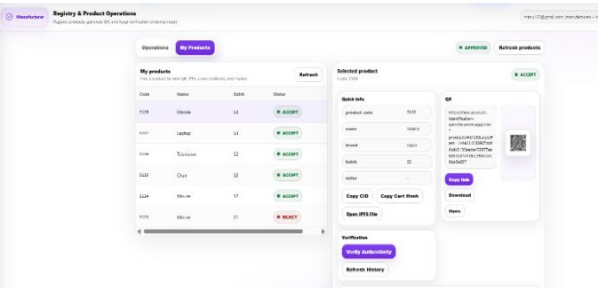

Fig -6: *Registered Products & QR Codes*

Product management dashboard showing registration, approval, and verification details.

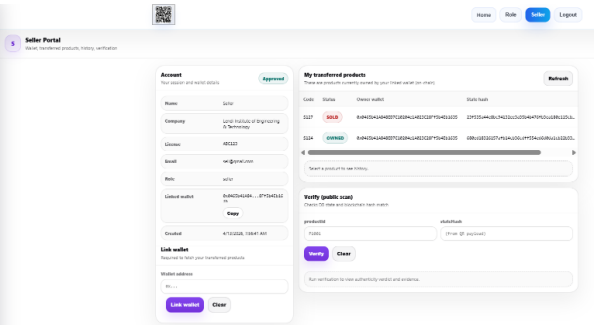

Fig-7: *Seller Dashboard & Ownership Transfer*

Seller dashboard for product management and verification in the authentication system.

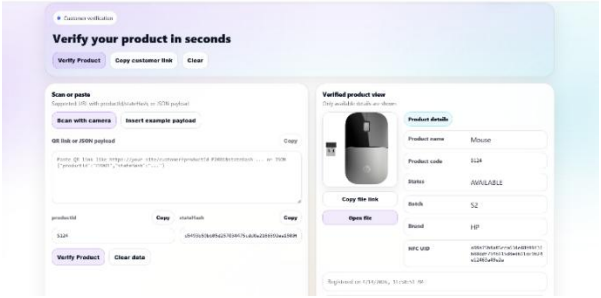

Fig-8: *Seller Dashboard & Ownership Transfer*

Customer verification interface for scanning QR codes and validating product authenticity

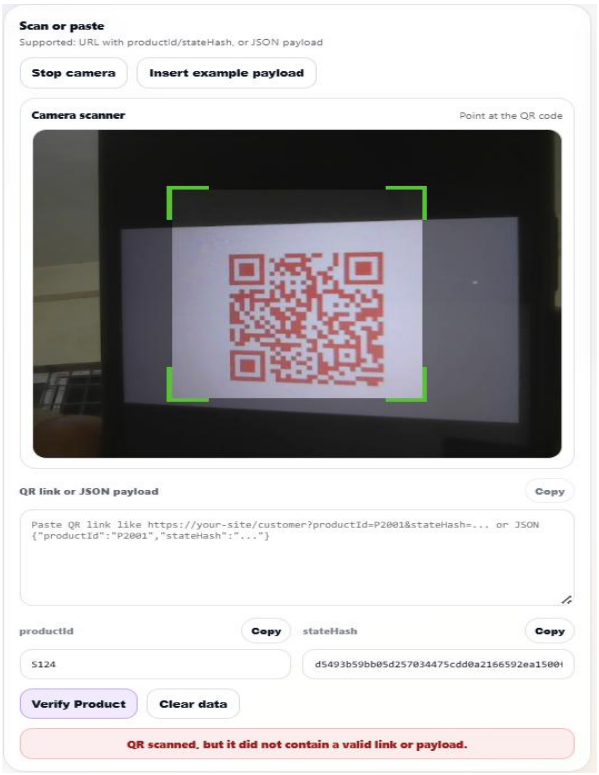

Fig-9: *QR Mismatch Detection*

Graphs

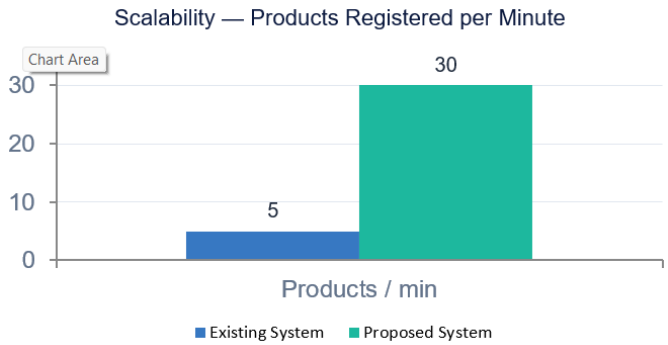

Fig -10: *Scalability*

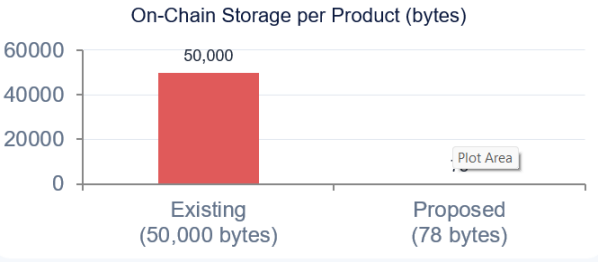

Fig -11: *Blockchain Storage*

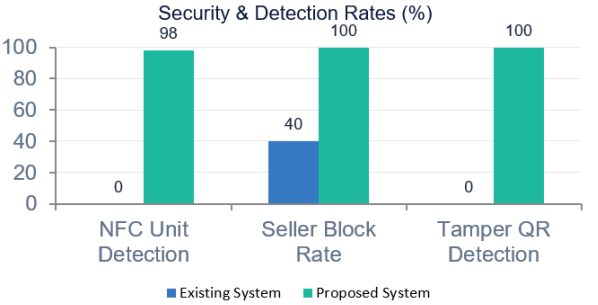

Fig -12: *Security*

VII. CONCLUSIONS

The Smart Product Authentication System using Blockchain with Cloud Integration provides an effective solution to address the growing problem of counterfeit products. By leveraging blockchain technology, the system ensures secure, transparent, and tamper-proof storage of product information. The integration of QR codes and NFC tags enables unique identification and real-time verification of products, improving trust among consumers and stakeholders. The use of cloud storage enhances the system's scalability by efficiently managing large volumes of product data, while blockchain maintains data integrity through secure hashing. The proposed system successfully overcomes the limitations of traditional authentication methods and provides a reliable framework for product verification. Overall, this approach contributes to improved supply chain transparency and helps in reducing the circulation of fake products.

VIII. FUTURE SCOPE

The future scope for the Smart Product Authentication System includes enhancement by developing a dedicated mobile application to provide easier access and real-time product verification for users. Advanced features such as AI-based counterfeit detection can be integrated to automatically identify suspicious or fake products. The system can also be extended to support real-time tracking of products across the supply chain, improving transparency, and monitoring. Additionally, it can be scaled for large-scale deployment across multiple industries and integrated with government or regulatory systems for better control and authentication standards.

REFERENCES

[1] N. Singh, B. Sahu, M. Lowanshi, and S. Verma, "Fake Product Identification Using Blockchain," *International Journal of Engineering Research and Development*, vol. 21, no. 2, pp. 309–316, 2025.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] J. Zhou, Y. Zhang, and X. Chen, "QR Code-Based Product Authentication System and Its Security Analysis," International Journal of Information Security and Applications, 2019.

[4] A. Juels, "RFID Security and Privacy: A Research Survey," IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 381–394, Feb. 2006.

[5] J. Benet, "IPFS – Content Addressed, Versioned, P2P File System," arXiv preprint arXiv:1407.3561, 2014. [Online]. Available: https://arxiv.org/abs/1407.3561

[6] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum White Paper, 2013. [Online]. Available: https://ethereum.org/en/whitepaper/

[7] K. Wasnik et al., "Detection of Counterfeit Products Using Blockchain," ITM Web of Conferences, vol. 44, 2022.

[8] R. Jadhav et al., "System for Identifying Fake Products Using Blockchain Technology," in Proc. 7th Int. Conf. Communication and Electronics Systems (ICCES), IEEE, 2022.