International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Detection of Product Forgery Using Blockchain

B M Prajwal[1], Mayur S[2], Kumarswamy S[3], Shobitha M[4]

[1,2,3,4]Student, Department of Computer Science, REVA University

Abstract

Supply chain management faces persistent challenges such as inefficiencies, poor coordination, and lack of transparency, with product counterfeiting emerging as a critical threat. Counterfeit goods, often indistinguishable from genuine products, undermine brand integrity and consumer trust. While existing solutions like RFID tags, AI-based detection, and QR codes offer some protection, they suffer from significant drawbacks—QR codes can be duplicated, and AI methods like CNNs require heavy computational resources. To address these limitations, this project proposes a blockchain-based system for secure product authentication and supply chain traceability. Blockchain technology provides a decentralized, tamper-proof ledger that records every transaction and movement of a product, ensuring transparency and immutability. Since data modification requires consensus across the network, the system offers robust protection against fraud and unauthorized alterations. By enabling real-time verification of a product's origin and journey, stakeholders can effectively identify and eliminate counterfeit items. Additionally, the decentralized nature of blockchain facilitates seamless data sharing among multiple parties, enhancing supply chain coordination without compromising security. This paper explores the implementation of blockchain for counterfeit detection, demonstrating its advantages over traditional methods in terms of reliability, scalability, and resistance to manipulation. The proposed solution not only mitigates counterfeit risks but also improves overall supply chain efficiency by fostering trust and accountability among manufacturers, distributors, and consumers. Ultimately, this research highlights blockchain's potential to revolutionize anti-counterfeiting measures while addressing key supply chain vulnerabilities.

Keywords: Blockchain Technology, Supply Chain Transparency,Product Authentication, Anti Counterfeiting System, Supply Chain Security, Consumer Protection, Brand Integrity, Product Traceability.

1. INTRODUCTION

Product counterfeiting occurs when a product is falsely presented as another brand's genuine item. It is a form of consumer fraud, typically defined as deceptive practices that result in financial or other losses for consumers. According to the Authentication Solution Providers' Association, counterfeiting costs the Indian economy INR 1 trillion annually, with incidents rising by an average of 20% from 2018 to 2020 [1]. Common counterfeit products include handbags, clothing, cosmetics, and electronics. Counterfeiting impacts not only the economy but also public health and safety [1, 3]. Low-quality products can cause significant harm such as skin diseases or electronic malfunctions [1].

Counterfeiting also damages brand reputations. Many customers unknowingly buy counterfeit items, and when these products fail to meet expectations, the legitimate companies suffer reputational damage. Customers often demand compensation from the authentic brand, placing businesses in a difficult position

[3, 14]. Counterfeit items erode trust among stakeholders, who may lose confidence due to counterfeiters' activities [2, 3].

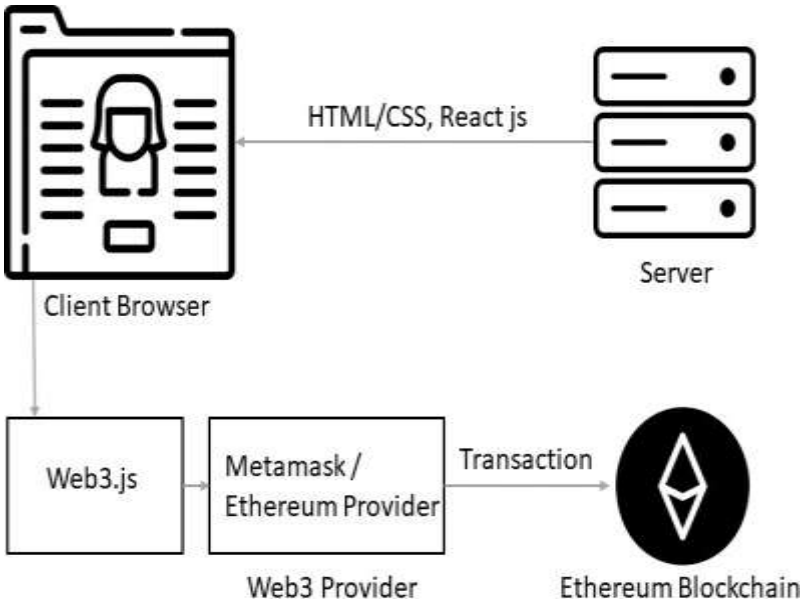
Effective strategies to address counterfeiting in global supply chains include increasing network transparency, cost control, pre-supply evaluation, and supplier relationship management [3, 18]. This paper aims to introduce a blockchain-based system to counteract counterfeiting, providing end-users and suppliers with a secure means to trace a product's supply chain [5, 9]. The proposed system seeks to combat brand counterfeiting by enabling transparency and verification of product authenticity [16,19].

2. METHODOLOGY

The system architecture for the Detection of Counterfeit Products using Blockchain is designed to provide transparency and security through a decentralized application (dApp) powered by Ethereum blockchain technology. The architecture consists of several key components that work together to track, verify, and authenticate products in the supply chain. These components include Ethereum smart contracts [5,17], QR codes, Web3.js, and a SQL database for account verification.

In this system, each role plays a crucial part in ensuring that products are accurately tracked from their creation at the manufacturer's end to their delivery to the customer. The integration of Ethereum blockchain smart contracts ensures the data is immutable, transparent, and secure, eliminating the possibility of tampering. Let's break down the architecture further, detailing each component and role, as well as the system's functionality.

Figure 1: System Architecture



2.1.Blockchain as the Foundation Against Forgery

Blockchain technology serves as a powerful antidote to forgery and unauthorized data changes. It works by storing data in blocks that are cryptographically linked to one another, creating a tamper-evident structure. Once a block is confirmed by the network, it is virtually impossible to change without altering all subsequent blocks—a task that would require an enormous amount of computing power and network consensus.

In our implementaion, each interaction with a product—whether it's creation, transit, handoff, or

verification—is recorded as a blockchain transaction. These transactions are:

- Cryptographically signed using the user's private Ethereum key.
- Permanently recorded in a transparent ledger.
- Time-stamped and associated with the exact user and action.

Because this ledger is distributed, there is no single point of failure. Even if a node tries to falsify data, the discrepancy is caught instantly as it fails to match the consensus copy on other nodes.

This immutable architecture ensures that no actor—internal or external—can forge records or delete history without detection.

2.2.SQL Database for Account Verification

While blockchain ensures data integrity and transparency, a SQL database is used to manage user accounts and verify access to the system. The SQL database stores user credentials for manufacturers, suppliers, and customers, ensuring that only authorized participants can register products or update their status.

The database allows for account management and authentication, ensuring that the system's users are legitimate and have the appropriate permissions to interact with the blockchain. For example, the supplier can only update product location details if they are registered and authenticated in the system [9,19].

2.3. The QR System

The implementation enables real-time authentication of products using easily accessible interfaces—primarily QR codes. QR codes are widely used due to their simplicity, low cost, and compatibility with nearly every smartphone. This ubiquity makes them an ideal tool for linking physical products to their digital records on the blockchain.

In this system, each product is assigned a unique identifier encoded into a QR code, which is physically affixed to the product packaging [13]. When scanned, the code retrieves the product's history from the blockchain and displays it to the user. The record includes all registered transactions: who created the product, who handled it during distribution, and its final destination [13, 15].

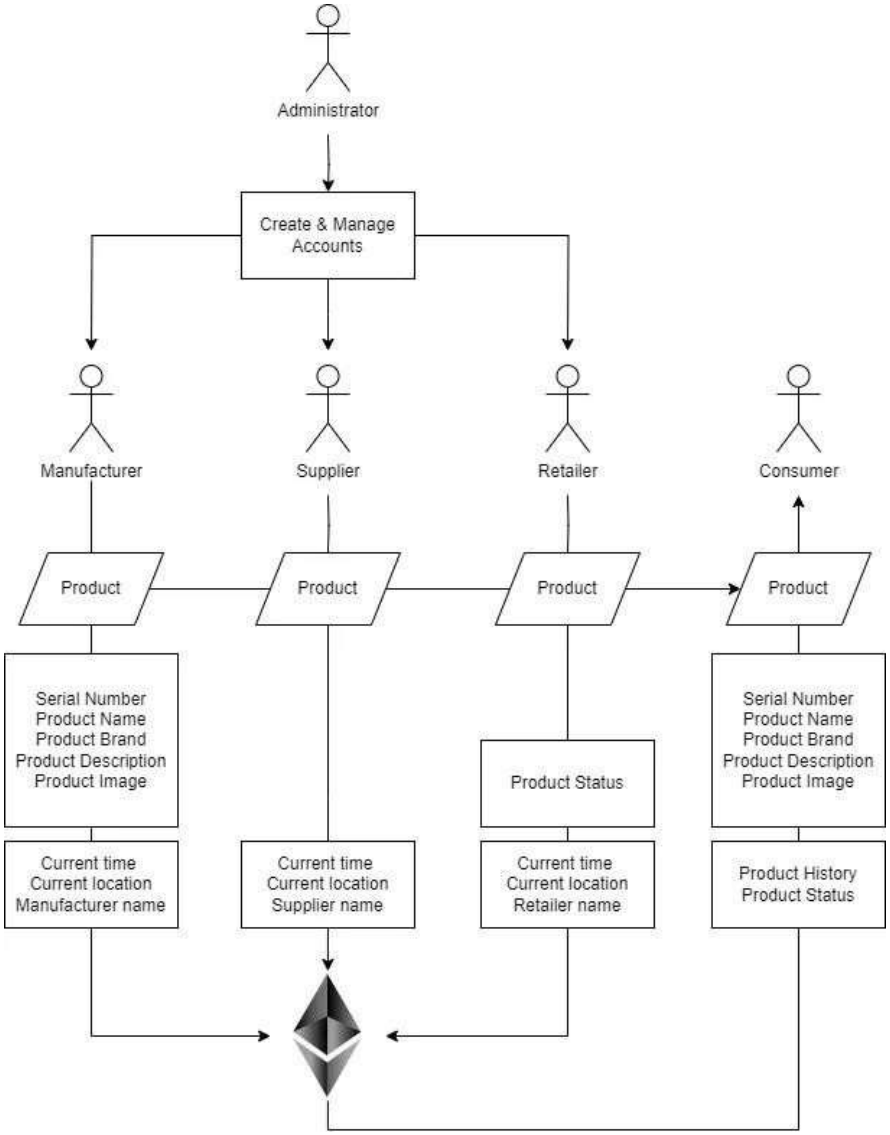
This enables:

- Manufacturers to associate new products with QR codes at the point of creation.
- Suppliers to update movement or handling data during distribution.
- Customers to verify a product's authenticity before or after purchase.

Real-time authentication via QR codes empowers consumers by allowing them to confirm the legitimacy of the products they buy. If the QR code has been tampered with, copied, or linked to inconsistent supply chain data, the system can immediately raise red flags, alerting users to the possibility of counterfeiting. This instant verification process builds consumer confidence, discourages the circulation of fake goods, and reduces dependency on third-party validators or authorities.

3. Modelling and Analysis

Figure 2: Detailed system model



3.1 Implementation Flow

- **Manufacturer** logs in → Enters product details → Generates QR code → Adds to blockchain via Metamask [17].
- **Supplier** scans QR → Adds transit data → Confirms via Metamask → Blockchain updated.
- **Customer** scans QR → Fetches full product history from blockchain → Verifies authenticity [5,17].

Step-by-Step Process:

- **Manufacturer Login:**

The manufacturer logs into the system through a secure authentication process. This may involve using a web interface where the manufacturer connects their Metamask wallet to authenticate their identity. The wallet acts as the manufacturer's unique identifier and provides authorization to interact with the Ethereum blockchain. Once logged in, the manufacturer is prompted to enter essential product details into the system. These details can include the product's name, description, source (where the product was made), and destination (where it is being shipped). Additionally, the manufacturer may input any additional

information that helps uniquely identify the product, such as serial numbers, batch codes, or custom identifiers. After entering the product details, the manufacturer's interface generates a unique **QR code** for the product. The QR code includes encrypted information that links directly to the product's details stored on the blockchain. This QR code serves as the key to access the product's history, allowing stakeholders in the supply chain to verify its authenticity and trace its journey. Once the product information is entered and the QR code is generated, the manufacturer can confirm and submit the product's details to the **Ethereum blockchain**. Using the Metamask wallet, the manufacturer initiates a transaction to store the product details in a smart contract on the Ethereum blockchain. The smart contract ensures that the product's data is immutable and can be updated only by authorized entities. The **Metamask wallet** serves as a bridge between the manufacturer's local interface and the Ethereum network. By confirming the transaction through Metamask, the manufacturer securely stores the product's information on the blockchain, ensuring that the data cannot be altered later.

- **Supplier** :

The supplier receives the product and scans the QR code generated by the manufacturer. The QR code links directly to the product's details stored on the blockchain, providing the supplier with essential product information such as its origin, manufacturer, and any prior transit data. The supplier can add transit-related information about the product, including location updates, shipping statuses, or any important logistics details. For example, the supplier may update the product's status from "Manufactured" to "In Transit" and include information about its current location or estimated arrival time. This data is important for tracking the product's movement and ensuring transparency at each stage of the supply chain. The supplier adds these updates through the system's user interface, which interacts with the blockchain. This ensures that all transit data is securely stored and associated with the product's record, making it accessible to anyone with the product's QR code. Similar to the manufacturer, the supplier needs to confirm the transaction using their **Metamask** wallet. When the supplier submits the updated transit data, the system initiates a transaction on the Ethereum blockchain. The supplier is prompted by Metamask to sign the transaction, which ensures that only authorized suppliers can update product data. This process guarantees that the updates are secure, verifiable, and traceable back to the supplier's wallet address.

- **Customer :**

When a customer receives a product, they can scan the QR code printed on the product's packaging or label using a mobile app or device. This QR code links to the product's unique ID stored on the blockchain. The system retrieves the product's details from the blockchain, providing a transparent and secure history of the product. After scanning the QR code, the customer's app or interface communicates with the Ethereum blockchain to fetch the product's full history. This history includes all updates made by the manufacturer and supplier, such as when the product was manufactured, where it was produced, where it has been shipped, and when it was last updated. This full product history is crucial for ensuring that the customer is purchasing an authentic product and not a counterfeit. The app may display the product's entire journey, highlighting key events like when it left the manufacturer, passed through the supplier, and reached the retailer. By reviewing the product's complete history, the customer can **verify the authenticity** of the product. For example, they can confirm that the product came from the expected source and that the shipping details match the product's physical characteristics (e.g., the product's serial number or model).

3.2 Role-Based Access Control and Wallet Mapping

A critical layer of protection in the system is the use of **role-based access control (RBAC)**, reinforced by

wallet address mapping. Unlike traditional login systems that can be bypassed or hacked, Ethereum-based access control uses **cryptographic wallet addresses** as identities [17].

Each user in the system—manufacturer, supplier, retailer, or customer—is linked to a unique wallet address. This address serves as their digital fingerprint. Smart contracts check this fingerprint before allowing any operation, such as:

- Registering a new product.
- Adding shipping or location data.
- Viewing or verifying product history.

This approach means that:

- **Only verified manufacturers** can create product entries.
- **Suppliers cannot manipulate manufacturer data**—they can only add their own tracking information.
- **Customers can read but not write to the blockchain**, ensuring a strict separation of privileges.

Unauthorized access becomes infeasible because no one can impersonate a wallet without the private key which is never stored or shared, but managed securely through platforms like Metamask [19]**.**

3.3 ProductTracker in Solidity

The **ProductTracker** smart contract is designed to enable the tracking and management of product information on the Ethereum blockchain. Using Solidity, the Ethereum smart contract programming language, this contract serves as the backbone of a decentralized application (dApp) that tracks product history in the supply chain. The contract allows manufacturers, suppliers, and other stakeholders to add and retrieve information about a product, ensuring transparency and traceability [15].

The contract enables products to be identified by a unique ID, and it tracks their journey from source to destination, including relevant details such as the product's name, source, destination, the address of the entity adding information, and the timestamp of the update. This information is stored on the blockchain, which is immutable and transparent, ensuring that no one can alter the data once it is added [15,19].

The core features of this smart contract include adding product history, retrieving product history, and managing product-related data using blockchain's decentralized nature.

3.4 Defending Against QR Code Forgery

QR code duplication is one of the most common ways counterfeiters try to trick verification systems. A malicious actor may copy a genuine QR code and apply it to fake goods, misleading customers and systems alike.

To counter this, we use several defense mechanisms:

- **One-to-One Mapping**: Each QR code maps to a unique product ID stored on the blockchain.
- **State-Linked Status**: When a QR code is scanned, the product's blockchain history is retrieved and verified. If the product shows a status or location mismatch, the system flags it.
- **Scan Event Logging**: The system can log metadata about the scan (e.g., IP address, timestamp, GPS location) to help trace misuse.
- **Expired State Logic**: Products marked as "sold," "expired," or "recalled" can trigger warnings if scanned again.

These measures create a multi-layered authentication system, making simple QR cloning ineffective.

4. Conclusion

In conclusion, this project successfully presents an innovative and robust solution to combat product for

gery by leveraging blockchain technology within the supply chain. The implementation allows manufacturers, suppliers, and retailers to securely contribute transaction details independently to the blockchain without altering other stakeholders' blocks. Solidity-based Ethereum smart contracts effectively ensure the immutability, authenticity, and robustness of each transaction.

For practical deployment, the project utilized local testing through Ganache with configurations set up at host 127.0.0.1 and port 7545 in the truffle-config.js file. Contracts were systematically compiled and deployed onto the Ethereum blockchain network using Truffle, following structured migrations. These migrations streamlined the contract deployment operations reliably.

The interface was developed using React, providing a seamless user experience. Web3.js facilitated smooth and secure integration between the React frontend and Ethereum blockchain, enabling essential blockchain operations such as sending ether, confirming transactions, and interacting with smart contract data. Wallet functionality within the blockchain environment was ensured by using Phantom Wallet—in conjunction with MetaMask accounts imported from Ganache—which seamlessly handled account authentication and transaction signing through secure confirmations.

The proposed solution effectively implements role-based access control, assigning distinct privileges to manufacturers, suppliers, and retailers. Manufacturers can securely add unique product identifiers and generate QR codes. Suppliers can update product-related information based on these QR codes, confirming additions securely via their wallets. Retailers perform final verification scans upon receiving goods, thereafter updating the product status to 'sold' or 'available'. The securely recorded blockchain ledger thereby allows customers to independently verify product authenticity and supply chain history, significantly reducing risks associated with counterfeiting.

The thorough deployment on a local test environment has validated crucial aspects of the system, including security, scalability, and reliability. The architecture and deployment process ensure system resistance against unauthorized data manipulation, as blockchain data updates require cryptographic wallet confirmation, ensuring considerable security against potential counterfeits and fraud.

Future expansions, such as developing a comprehensive mobile application available on both iOS and Android, will facilitate wider accessibility and interaction with data through intuitive QR scans and real-time blockchain insights. Additional enhancements—like in-app notifications for supply chain updates, product recalls, wallet-based user authentication, and deeper blockchain integration—will enable users to authenticate products swiftly and securely directly from their mobile devices, thereby broadly increasing consumer confidence and brand reputation protection.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. 1. ASPA, The state of counterfeiting in india 2021.
2. Y. Lu, Journal of Management Analytics 5, 1 (2018)
3. F. Casino, T.K. Dasaklis, C. Patsakis, Telematics Informatics 36, 55 (2019)
4. M. Peck, IEEE Spectrum 54, 26 (2017)
5. S. Idrees, M. Nowostawski, R. Jameel, A. Mourya, Electronics 10, 951 (2021)
6. Zignuts Technolab, How blockchain architecture works? basic understanding of blockchain and its architecture,2022.
7. J. Ma, S.Y. Lin, X. Chen, H.M. Sun, Y.C. Chen, H. Wang, IEEE Access 8, 77642 (2020)
8. M.J.L.I.N.M. J.M. Bohli, N. Gruschka, IEEE 10, 9 (2013)

9. C. Shaik, Computer Science & Engineering: An International Journal (CSEIJ) 11 (2021)
10. M.A. Benatia, D. Baudry, A. Louis, Journal of Ambient Intelligence and Humanized Computing pp. 1– 10 (2020)
11. G. Khalil, R. Doss, M. Chowdhury, IEEE Access 8, 47952 (2020)
12. M.A. Habib, M.B. Sardar, S. Jabbar, C.N. Faisal, N. Mahmood, M. Ahmad, Blockchain-based supply chain for the automation of transaction process: Case study based validation, in 2020 International Conference on Engineering and Emerging Technologies (ICEET) (IEEE, 2020), pp. 1–7
13. E. Daoud, D. Vu, H. Nguyen, M. Gaedke, Improving Fake Product Detection Using Ai-Based Technology, in 18th International Conference e-Society (2020)
14. S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, J. Zhang, A blockchain-based supply chain quality management framework, in 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE) (IEEE, 2017), pp. 172–176
15. K. Toyoda, P.T. Mathiopoulos, I. Sasase, T. Ohtsuki, IEEE access 5, 17465 (2017)
16. M. Nakasumi, Information sharing for supply chain management based on block chain technology, in 2017 IEEE 19th conference on business informatics (CBI) (IEEE, 2017), Vol. 1, pp. 140–149
17. G. Wood et al., Ethereum project yellow paper 151, 1 (2014)
18. A. Ghadge, A. Duck, M. Er, N. Caldwell, Supply Chain Forum: An International Journal 22, 87 (2021)
19. I. Singhal, International Journal for Research in Applied Science and Engineering Technology 9, 291(2021)