BLOCKCHAIN TECHNOLOGY

Assignment - 4

Implementation and Security Analysis of a Blockchain-Based Supply Chain Using Hyperledger Fabric

Abstract

Blockchain technology provides an immutable and decentralized framework for ensuring data integrity and transparency across distributed systems. This report presents the design, deployment, and analysis of a blockchain-based supply chain using Hyperledger Fabric. The implementation demonstrates a permissioned network connecting three organizations—Manufacturer, Distributor, and Retailer—across public and private channels. The system tracks product creation, shipment, and receipt while enforcing strict access control through Fabric's Membership Service Provider (MSP). Furthermore, the report includes a security threat analysis for different blockchain layers and suggests mitigation techniques. The findings show that a well-configured Fabric network can ensure confidentiality, availability, and integrity of supply chain data, thus reducing fraud and unauthorized manipulation.

Introduction

Blockchain technology has revolutionized supply chain management by introducing transparency, traceability, and immutability to product transactions. Traditional supply chain systems often suffer from inefficiencies such as lack of visibility, counterfeit products, and data tampering. In this project, Hyperledger Fabric, a permissioned blockchain framework, is used to build a secure and transparent network that connects manufacturers, distributors, and retailers. Each participant interacts with the ledger through chaincode (smart contracts) to perform operations like shipment creation, transfer, and verification. Furthermore, this report explores potential blockchain security vulnerabilities and mitigation strategies to ensure the resilience of such systems in real-world applications.

Task 1: Blockchain-Based Supply Chain Implementation

The objective of Task 1 was to develop a permissioned blockchain network using Hyperledger Fabric that enables decentralized supply chain tracking. Three organizations

were configured to represent key participants: Org1 (Manufacturer), Org2 (Distributor), and Org3 (Retailer). The network uses two channels: a public channel named 'main-supply' accessible by all organizations, and a private channel 'manu-dist' restricted to Org1 and Org2 for confidential shipment operations.

The network setup was performed on Ubuntu 22.04 with Docker Engine and Fabric v2.5 binaries [1]. Docker Compose orchestrated peer and orderer containers. The Fabric test network provided the base configuration for creating the orderer and peer organizations. The scripts directory contained helper scripts for network startup, channel creation, and chaincode deployment. Key scripts included 'start_testnet.sh' to initialize the orderer and peers, 'create_manu_dist.sh' to create the private channel, and 'deploy_cc_main.sh' and 'deploy_cc_manudist.sh' for chaincode lifecycle management.

Table 1 summarizes the main network components used in the implementation.

Table 1 Summary of main network components

Component	Description
Org1 (Manufacturer)	Creates products and initiates shipment records.
Org2 (Wholesaler)	Receives shipments and updates product ownership.
Org3 (Retailer)	Final receiver of delivered goods; performs product verification.

The Fabric network was successfully deployed and configured using the provided Docker Compose scripts. Each participant organization had a peer node and certificate authority, ensuring identity management through digital certificates. The following steps summarize the core implementation process:

1. The manufacturer created a new shipment record containing product details such as product ID, batch number, and timestamp. The event was logged on the ledger and made visible only to authorized participants on the private channel.
2. The distributor received the shipment from the manufacturer. Upon verification, the chaincode executed a transfer function that updated the product's state in the ledger. A corresponding event was recorded to ensure traceability.
3. The retailer acknowledged the receipt of the product, marking the final transaction in the supply chain lifecycle. Ledger queries confirmed the immutability of all recorded events.
4. control was verified using Fabric's Certificate Authority. Unauthorized participants were unable to read or modify restricted data. Consensus mechanisms

ensured that all transactions were endorsed and validated before committing to the ledger.

Once the network was live, the chaincode 'trackchain-js' was deployed to both channels. The chaincode, written in JavaScript, managed asset creation, state transitions, and access control. The main functions implemented were CreateProduct(), CreateShipment(), ReceiveShipment(), GetProduct(), and GetProductHistory(). Each transaction was endorsed according to channel-specific policies—on the 'main-supply' channel, endorsements from any organization were accepted, while on 'manu-dist', endorsements required Org1 and Org2 signatures. This configuration ensured that only relevant participants could modify or view confidential data [2].

The following simulated transaction sequence was executed and verified via peer CLI commands:

[Org1] → peer chaincode invoke ... Args:[CreateProduct,P200,SKU-ALPHA, Smart Sensor Module]
[Org1] → peer chaincode invoke ... Args:[CreateShipment,S200,P100,Org2MSP]
[Org2] → peer chaincode invoke ... Args:[ReceiveShipment,S100]
[Org3] → peer chaincode invoke ... Args:[ReceiveShipment,S200] → Authorization denied
[Query] → peer chaincode query ... Args:[GetProduct,P200]
[Query] → peer chaincode query ... Args:[GetProductHistory,P200]

The log outputs validated that the product's lifecycle—from creation to delivery—was recorded immutably across the ledger. Unauthorized access attempts by Org3 were rejected by the chaincode authorization logic, demonstrating correct MSP-based enforcement. The 'GetProductHistory' call displayed all state transitions including transaction IDs and timestamps.

The network utilized TLS encryption for all peer–orderer communications, ensuring data confidentiality. Chaincode lifecycle endorsement policies prevented unapproved code from being deployed. Moreover, Fabric's certificate authority (CA) issued cryptographic identities to enforce organizational roles and prevent spoofing. These configurations collectively established trust between participants and ensured consistent ledger updates [3].

Task 2: Security Threat Analysis and Mitigation Strategies

Task 2 involved identifying potential security vulnerabilities within different layers of the blockchain architecture—network, consensus, transaction, and application—and proposing suitable mitigation strategies. The evaluation was performed using a structured threat modeling approach inspired by STRIDE and blockchain-specific frameworks [4].

Table 2 lists identified threats and their mitigation mechanisms.

Table 2 List of identified threats and their mitigation mechanisms

Threat	Impact	Mitigation Strategy
Double Spending / Replay Attacks	Inconsistent ledger state or duplicate transactions.	Fabric's MVCC validation and unique transaction IDs prevent replayed invocations.
Sybil Attack	Malicious node impersonation in network consensus.	Permissioned membership via MSP ensures identity-based participation.
Smart Contract Exploits	Malicious code injection or logic errors in chaincode.	Peer endorsement policies, code audits, and chaincode approvals mitigate risk.
Man-in-the-Middle (MITM) Attack	Intercepted peer–orderer communications leading to spoofing.	TLS encryption and mutual authentication enforced between all components.

Threat modeling tools such as Microsoft Threat Modeling Tool and OWASP recommendations were conceptually applied to visualize data flow diagrams and identify trust boundaries. The analysis confirmed that Fabric's channel architecture and endorsement policies serve as key mitigations against most network and transaction-level threats [5].

An example of a potential attack scenario involved an unauthorized entity attempting to execute 'ReceiveShipment' for a shipment not addressed to its MSP. This attack was successfully prevented by the chaincode's runtime MSP verification, which cross-checked the invoker's certificate organization. The peer CLI returned an error message: 'Only the intended receiver can accept this shipment.' This outcome validated Fabric's fine-grained access control and chaincode-level defense mechanisms.

Recommendations for Secure Deployment

To enhance resilience, several measures were recommended:

- Employ multi-node Raft consensus for fault tolerance.
- Use Hardware Security Modules (HSMs) to protect private keys.
- Enable periodic certificate rotation through Fabric CA.
- Define Private Data Collections (PDCs) for high-confidentiality records.
- Conduct regular chaincode audits and runtime monitoring.
- Restrict admin-level operations and enforce principle of least privilege.

Conclusion

The project successfully demonstrated the feasibility of integrating Hyperledger Fabric into supply chain workflows. The multi-organization setup proved effective for transparent and tamper-proof tracking of goods. Access control and endorsement policies ensured data integrity and accountability among participants. The accompanying security analysis revealed Fabric's robust defenses against major blockchain threats while emphasizing the need for operational best practices. This study validates blockchain's capability to revolutionize enterprise supply chains by providing verifiable, traceable, and secure data exchange mechanisms.

References

[1] Hyperledger Foundation, "Hyperledger Fabric v2.5 Documentation," 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/

[2] M. Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *EuroSys Conference*, 2018.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[4] OWASP Foundation, "Blockchain Security Threat Modeling Guide," 2023.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, 2020.

Name- Eva Hemantkumar Shah
Registration No.- 220911668