IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 5, November 2024

Blockchain based Intelligent Transportation System (ITS)

Dr. Sonali Ridhorkar, Sejal Raghorte, Suraj Dudhe, Vaishnavi Bais,
Vigyat Singh, Swikruti Nandurkar and Yuvraj Singh
Department of Computer Science and Engineering
G. H. Raisoni College of Engineering and Management, Nagpur, Maharashtra, India

Abstract*: A blockchain-driven system is designed for secure, decentralized data storage in Intelligent Transportation Systems (ITS). It integrates Vehicle-to-Infrastructure (V2I) communication, aimed at improving road safety, traffic control, and vehicle traceability while offering real-time insights to support better planning decisions. By employing a cross-chain communication framework, the system allows for seamless interoperability between different blockchains through the use of relay nodes. Identity-based encryption safeguards node participation, while a standardized protocol enables smooth communication and transactions, promoting trust among users. This system, developed with Hyperledger Fabric, ensures the integrity and privacy of data across varied blockchain environments within smart city transportation networks.*

Through the decentralized nature of blockchain technology, the system minimizes the risks of data tampering and unauthorized access, delivering robust solutions for handling vehicle information. Furthermore, it enhances vehicle traceability and transparency of vehicle movements, granting secure access to transportation data for all stakeholders. This contributes to optimized traffic flow, a reduction in congestion, and fosters the development of a more resilient and responsive transportation infrastructure.

Keywords: blockchain-driven system

I. INTRODUCTION

The rapid advancement of intelligent transportation systems (ITS) has enabled efficient management of urban traffic, vehicle coordination, and enhanced road safety by leveraging data from multiple sources, such as vehicles, infrastructure, and government agencies [1]. However, as these systems evolve, the volume of sensitive data being transmitted over open networks has raised significant concerns regarding data security and privacy [2]. Traditional ITS architectures rely heavily on centralized systems for data processing, which introduces vulnerabilities in terms of trust, data integrity, and cross-institutional collaboration [3].

Blockchain technology, specifically Hyperledger Fabric, offers a promising solution to overcome these challenges by decentralizing the management of data within ITS [4]. Hyperledger Fabric provides a permissioned blockchain environment, enabling secure and private transactions between various stakeholders in the transportation ecosystem [5]. By employing a distributed ledger and consensus mechanisms, data within the system becomes immutable and traceable, ensuring that unauthorized tampering is nearly impossible [6]. This decentralization minimizes the need for third-party trust and enhances the integrity and security of transactions across multiple organizations [7].

The integration of blockchain into ITS not only enhances the security of data exchange but also facilitates smoother cross-domain interactions between various platforms and institutions [8]. Smart contracts, deployed on Hyperledger Fabric, ensure that data is automatically verified and validated across different nodes, reducing the risk of fraudulent activities and improving operational efficiency [9]. Additionally, this framework enables secure identity verification, access control, and data sharing without reliance on a single centralized entity, thereby addressing the challenges of scalability and trust in traditional systems [10].

This research proposes an ITS framework built on Hyperledger Fabric, focusing on the benefits of secure transactions, cross-chain data sharing, and privacy-preserving mechanisms [11]. The framework introduces innovations such as relay chain-gateway node models and IBE-based secure access controls, which together enhance interoperability and trust in

DOI: 10.48175/568

multi-organizational ITS environments [12]. The proposed solution has been tested using the NGSIM dataset on Hyperledger Fabric to evaluate its performance, highlighting its potential to revolutionize how data is managed and secured within the transportation industry [13].

II. RELATED WORK

The selected papers explore the use of blockchain and AI to improve Intelligent Transportation Systems (ITS). Blockchain is applied in areas like vehicle theft detection, secure data communication, and identity authentication, ensuring data integrity and tamper-proof storage. AI is used to optimize vehicle routing and improve energy efficiency in transportation systems, enhancing traffic flow and system performance.

[1]This reference introduces a blockchain-based framework for preventing vehicle theft. It uses smart contracts to automate the detection of theft attempts and trigger responses. The system ensures that vehicle ownership and control can be securely verified in a decentralized manner. Blockchain's tamper-proof nature helps to securely store vehicle data, and smart contracts allow for real-time theft detection, making this system highly secure and efficient.

[2]This paper focuses on an AI-driven routing protocol for Intelligent Transportation Systems (ITS). The protocol aims to optimize vehicle routing while minimizing energy consumption. It leverages artificial intelligence (AI) to enhance decision-making in routing, contributing to energy-efficient traffic management and improving the overall efficiency of transportation networks.

[3]This study presents a blockchain-based solution for IoT communication in the Internet of Vehicles (IoV), utilizing the Ethereum platform. It focuses on ensuring secure and decentralized communication between vehicles using smart contracts and distributed ledger technology. The system aims to address security vulnerabilities in traditional IoT-based vehicle communications, offering a more secure, transparent, and decentralized alternative for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication.

[4]This paper discusses a blockchain-based framework for identity authentication in vehicular networks. The proposed system ensures that vehicle identities are authenticated securely and can be revoked quickly if necessary. The use of blockchain provides a decentralized approach to managing identities, preventing malicious actors from exploiting the network and ensuring secure interactions between vehicles and the network.

These works highlight the transformative role of blockchain and AI in ITS, offering secure, decentralized solutions for vehicle communication, traffic management, and data validation. Together, these technologies promise to enhance the safety, efficiency, and scalability of smart transportation networks.

III. TECHNICAL BACKGROUND

The three technologies used in building the intelligent transportation system are

- Blockchain Interoperability
- Peer-To-Peer Network
- Smart Contracts

3.1 Blockchain Interoperability

Think of each blockchain as a separate country with its own language, rules, and systems. Now imagine you want to trade or share information between these countries—this is where **cross-chain communication** comes in.

Cross-chain communication enables different blockchains (which operate independently like different countries) to talk to each other and share value or data. For example, you might want to transfer assets from a Bitcoin blockchain to an Ethereum blockchain, or share important data between two blockchain platforms. Similarly in the intelligent transportation system the different blockchain networks of the infrastructure could easily communicate over the cross-chain.

Without cross-chain communication, this would be very difficult because each blockchain has its own rules. But with **cross-chain protocols**, you can build "bridges" that allow these blockchains to exchange information or assets securely.

How it works: A cross-chain transaction starts on a **source blockchain** (the sender) and is completed on a **target blockchain** (the receiver). It's a way to move assets or data from one chain to another without breaking their security rules.

Why it matters: This technology is critical for improving the interoperability of blockchains, allowing different blockchain ecosystems to work together. In the future, this could enable more complex blockchain applications, like combining the strengths of multiple blockchains in a single system.

3.2 Peer to Peer to Network

LibP2P is like a highly flexible and customizable postal service for computers, helping them communicate with each other directly, without the need for a central hub. It's used to build **peer-to-peer (P2P) networks**, where devices (or nodes) can find, communicate, and share data directly with each other. It's a bit like a social network where everyone is connected, and anyone can chat with anyone else without needing a central server (like a government or an internet provider) to manage the conversation.

libP2P started out as a networking layer for **IPFS (Inter Planetary File System)** but has since become the go-to tool for building decentralized applications and blockchain networks.

Key Features for building ITS:

- **Node Discovery**: Just like meeting new people, libP2P helps devices (nodes) find each other in a network.
- **Secure Communication**: It ensures the data sent between nodes is secure and private.
- **Multi-Protocol Support**: It can support many types of communication protocols, making it adaptable to different needs.
- **Scalability**: It's highly flexible, so it can be used in small applications or huge networks.
- **Why it's important**: libP2P allows decentralized applications to communicate without relying on a central authority. This is especially valuable for building blockchain systems, as it helps with the flow of information between nodes, making the system more robust and resistant to failures or attacks.

3.3 Smart Contracts

Imagine you and a friend bet ₹50 on a football game. You could hire a third person to hold the money and pay the winner, but this person might make mistakes or take a cut. Instead, you could use a **smart contract**. This is a computer program that automatically holds and releases the money based on the outcome of the game. The program checks the games result and automatically pays the winner, all without needing any middleman.

Smart contracts are self-executing agreements coded onto a blockchain. Once they are set up, they automatically perform actions when predefined conditions are met. The best part is that they don't require trust in a third party because the blockchain guarantees that the contract will execute exactly as programmed.

Types of Smart Contracts:

- **Deterministic Contracts**: These contracts rely only on the information already on the blockchain. They don't need external data to function.
- **Non-Deterministic Contracts**: These need data from outside the blockchain (like a football score). They rely on trusted external sources called **oracles** to get that information.

Popular Platforms:

Ethereum is the most popular platform for smart contracts, and it uses a programming language called **Solidity**.

Other blockchains like **Bitcoin**, **Hyperledger Fabric**, and **Polkadot** also support smart contracts but may use different languages or have different features.

Why they're useful:

- **Automation**: Smart contracts remove the need for human intermediaries, saving time and cost.
- **Security**: Since smart contracts run on a blockchain, they are secure and immutable (can't be changed once deployed).
- **Transparency**: Anyone can see the contract and its execution on a public blockchain, ensuring transparency.

DOI: 10.48175/568

IV. AN INTELLIGENT TRANSPORTATION SYSTEM BASED ON RELAY NODES USING A SECURE CROSS-CHAIN SOLUTION

A new cross-chain communication architecture has been introduced for heterogeneous Intelligent Transportation Systems (ITS), integrating institutional chains, facility chains, and relay node chains. The system uses relay nodes to facilitate secure data exchange across different blockchains, utilizing Identity-Based Encryption (IBE) along with cross-chain interaction mechanisms. Each organization connects to the cross-chain network through dedicated relay nodes, enabling secure cross-chain transactions while keeping non-cross-chain activities confined to their respective blockchains. This design strengthens interoperablity and enhances data security within the ITS ecosystem.

4.1 Cross-Chain Interaction Framework Using Relay Nodes for ITS

The cross-chain interaction framework outlined in this architecture involves multiple components that work together to ensure seamless and secure communication between blockchain networks in ITS. These components include transport entities, application blockchains, cross-chain contracts, relay nodes, and specialized communication protocols. The cross-chain interaction process allows for user-initiated transactions to flow between different blockchain ecosystems, with each step ensuring data integrity and compliance with security protocols.

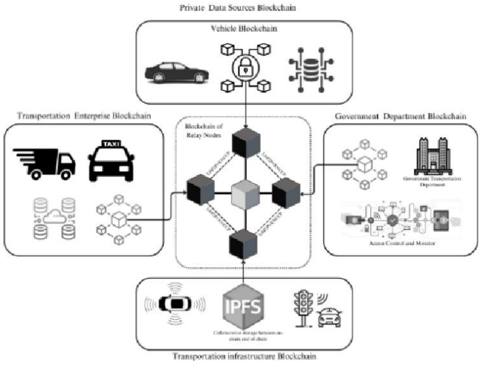

Private Data Sources Blockchain
Vehicle Blockchain
Transportation Enterprise Blockchain
Blockchain of Relay Nodes
Government Department Blockchain
TAXI
Government Transportation Department
Access Control and Monitor
IPFS
Collaborative storage between on-chain and off-chain
Transportation infrastructure Blockchain

Fig 1: Cross-Chain Communication Model for an Intelligent Transportation Network

It integrates different types of blockchains, each serving a specific function within ITS:

- **Traffic Data Sources Blockchain**: Collects real-time data from sensors and devices related to traffic flow, vehicle speeds, accidents, and road conditions.
- **Transportation Department Blockchain**: Acts as the central authority responsible for maintaining data integrity, managing access control, and facilitating data sharing among relevant departments.
- **Transport Enterprise Blockchain**: Focuses on managing logistics, including vehicle tracking, fleet management, route optimization, and transaction handling.
- **Transport Infrastructure Blockchain**: Stores and maintains data related to physical transportation infrastructure such as roads, bridges, and tunnels.

The architecture operates through the following processes:

- **Data Collection**: Traffic-related data is continuously collected from various sources across the transportation network and uploaded to the corresponding blockchain.
- **Data Sharing and Processing**: Relay nodes, utilizing the libP2P protocol and the Relay Node Cross-Chain Protocol (RNCCP), enable the seamless exchange of data between different blockchains.
- **Data Access and Regulation**: Authorized stakeholders can request access to data, with the Transportation Department Blockchain ensuring proper regulation and secure handling of information.
- **Node Authentication**: Identity-Based Encryption (IBE) is used to authenticate nodes within the network, ensuring that only verified entities can access sensitive data and preventing unauthorized access.
- **Cross-Chain Transactions**: Relay nodes facilitate secure and efficient transactions between disparate blockchains, enabling seamless interactions across the network.

- **Collaborative Data Storage:** The InterPlanetary File System (IPFS) is utilized to offer efficient storage solutions for managing both on-chain and off-chain data.

This cross-chain communication architecture offers a highly secure, transparent, and efficient data management framework, thereby improving decision-making, optimizing traffic management, and enhancing the overall quality of transportation services.

4.2 Relay Node-Based Cross-Chain Interaction Model for Intelligent Transportation Systems:

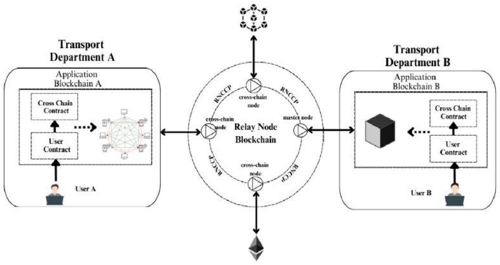


Fig 2: Cross-Network Architecture Leveraging Relay Nodes

The relay node-based cross-chain interaction model is designed to facilitate secure and efficient communication between diverse blockchain networks within the Intelligent Transportation System (ITS) ecosystem. It relies on key components such as transport entities, application blockchains, cross-chain smart contracts, relay nodes, and dedicated communication protocols to enable seamless data exchange and secure transaction processing across multiple blockchains.

This model includes the following essential processes:

- **Initiation of Cross-Chain Transactions**: Users initiate a cross-chain transaction from one blockchain within the ITS environment, intending to interact with another blockchain.
- **Interaction with Cross-Chain Smart Contracts**: Once the transaction is initiated, the user's smart contract communicates with a corresponding cross-chain contract. This contract acts as a bridge between the different blockchains involved in the transaction.
- **Secure Transaction Transmission through Relay Nodes:** The relay node linked to the originating blockchain obtains the transaction request and sends it to the relay node chain, which manages the cross-chain interaction. This process guarantees the secure transfer of the transaction from one blockchain to another.
- **Transaction Processing by the Relay Node Chain**: The relay node chain is responsible for validating transactions, ensuring adherence to the rules and protocols of both the source and destination blockchains. It processes the data by verifying its authenticity and integrity before allowing the transaction to proceed.
- **Data Exchange Across Blockchains**: Relay nodes handle the exchange of data between the source and destination blockchains. They ensure that the transaction data is securely transferred and properly interpreted across both networks.
- **Completion of the Cross-Chain Transaction**: After successful data exchange and validation, the transaction is executed on the destination blockchain. The result of the transaction is then communicated back to the initiating user via the relay nodes, completing the cross-chain process.

This cross-chain interaction model significantly enhances **interoperability** by allowing different blockchain systems to communicate and collaborate. It improves **scalability** by efficiently handling a high volume of transactions across diverse networks. Furthermore, it strengthens **security** by employing trusted relay nodes and rigorous validation mechanisms to ensure the integrity of data and compliance with blockchain protocols. By optimizing these factors, the model substantially improves the overall efficiency and reliability of ITS operations

4.3 Secure Data Transmission in ITS Relay Nodes Through Identity-Based Encryption

The use of Identity-Based Encryption (IBE) offers a streamlined approach to securing relay node communication within Intelligent Transportation Systems (ITS). First introduced by Shamir in 1985, Identity-Based Encryption (IBE)

DOI: 10.48175/568

streamlines the traditional Public Key Infrastructure (PKI) by allowing a user's identity, such as an email or network ID, to serve as their public key. This method streamlines the process by removing the need for complex certificate management systems. In 2001, Boneh and Franklin developed the first practical IBE scheme utilizing bilinear pairings on elliptic curves, which has since paved the way for more efficient cryptographic solutions.

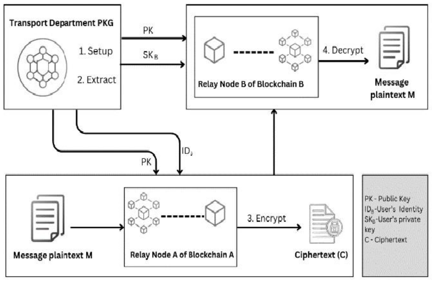


Fig 3: Process Flow for Encryption and Decryption Using IBE

In the context of ITS, IBE strengthens the security of cross-chain communication by allowing relay nodes to authenticate and exchange data without relying on conventional certificates, thus reducing overhead and improving system efficiency. The IBE framework is built around the following key processes:

1. **System Initialization (Setup):** The IBE system is initialized by generating a set of public parameters (PK) that are accessible to all entities in the network, along with a master secret key (MSK) that is known only to a trusted authority, typically referred to as the Private Key Generator (PKG). These parameters are essential for the cryptographic operations performed throughout the system.
2. **Private Key Generation (Extract):** When a user or relay node within the ITS requests their private key, the PKG uses the master secret key (MSK) and the user's unique identity (ID) to generate a private key (SKID). This private key is specific to the user's identity, enabling secure communication tailored to that individual entity without the need for traditional certificate authorities.
3. **Message Encryption (Encrypt):** To ensure secure communication between relay nodes or other entities within the ITS, messages are encrypted using the recipient's identity as the public key. This process generates ciphertext (C), which can only be decrypted by the intended recipient, ensuring confidentiality and preventing unauthorized access to sensitive information.
4. **Message Decryption (Decrypt):** To decrypt a message, the recipient uses their private key (SKID) in combination with the public parameters (PK) to unlock the ciphertext (C) and retrieve the original message (M). This ensures that only authorized users, whose identities have been authenticated through the IBE system, can access the content of the communication.

By leveraging IBE, the ITS architecture ensures that communication between relay nodes and other network participants is secure and efficient. Unlike traditional PKI systems, IBE eliminates the need for extensive certificate management, reducing operational complexity and enhancing the overall security of data transmission within the ITS ecosystem. This cryptographic approach also provides scalability, making it well-suited for the increasingly interconnected and decentralized nature of ITS networks.

4.4 Cross-Chain Communication Mechanism for Intelligent Transportation Systems

The cross-chain communication mechanism plays a critical role in enabling secure and seamless data exchange across different blockchain networks within an Intelligent Transportation System (ITS). This process ensures that various blockchain systems, such as those managing traffic data, infrastructure, and transportation enterprises, can communicate and collaborate effectively. Through this mechanism, cross-chain interactions facilitate asset transfers, information exchange, and collaborative decision-making, improving overall ITS functionality.The cross-chain interaction process is comprised of the following steps:

1. **Transaction Initiation:** The process begins when a user on one blockchain initiates a transaction that requires data or asset exchange with another blockchain within the ITS. The transaction could involve various

DOI: 10.48175/568

operations, such as requesting real-time traffic data, transferring assets between entities, or initiating contracts across different transportation networks.

2. **Interaction with Cross-Chain Smart Contracts:** Upon initiating the transaction, the user's contract communicates with a cross-chain smart contract within the source blockchain. This contract acts as a bridge, facilitating communication between the source blockchain and other blockchains involved in the transaction.
3. **Transmission via Relay Node Chain:** The cross-chain contract forwards the transaction request to the relay node chain, a specialized network of nodes responsible for managing cross-chain communication. These relay nodes validate the transaction and transmit it to the appropriate blockchain, ensuring that all necessary security protocols are observed during the process.
4. **Processing on the Destination Blockchain:** Once the relay node chain has successfully transmitted the transaction, the relay node on the destination blockchain retrieves the requested data or facilitates the required asset transfer. This step ensures that the operation follows the protocols and guidelines set by the destination blockchain, making sure everything proceeds correctly within that network.
5. **Data Relay and Response Transmission:** After processing the transaction on the destination blockchain, the relay node transmits the retrieved data or the result of the transaction back through the relay node chain. This ensures the secure transmission of data across blockchains while maintaining the integrity and authenticity of the transaction.
6. **Final Delivery to Source Blockchain:** The relay node chain transmits the retrieved information or transaction outcome back to the cross-chain contract on the source blockchain. The cross-chain contract then delivers the data or result to the user’s contract, completing the cross-chain interaction process.

This cross-chain communication mechanism enhances interoperability across different ITS blockchains, allowing diverse blockchain systems to work together seamlessly. By enabling secure and efficient data sharing, this mechanism facilitates better decision-making, resource optimization, and real-time responses in transportation systems. Moreover, the relay node architecture ensures the authenticity and security of each cross-chain interaction, protecting sensitive information and maintaining the integrity of the ITS network.

V. METHODOLOGY

5.1 Interaction between smart-contracts and cross-chain

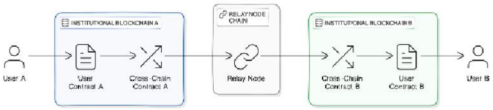


Fig 6: Cross-chain Interaction Structure

The smart contract outlined here serves as a connector for various blockchains, enabling them to interact with one another. It features a primary storage contract that outlines key events and parameters necessary for cross-chain transactions. There are two main reference contracts: one for overseeing relay node data and blockchain identifiers (ADD), and the other for managing string data operations (STR-OPE).

The procedure starts when a user initiates a cross-chain transaction through the startCrossChainTx() function. This function determines the transaction type, generates a transaction object, and issues a notification event for relay nodes to process. Relay nodes then utilize the sendAckedTx() function to verify they received the transaction, checking their authorization and confirming the contract version is accurate. Based on the transaction's status, they revise the transaction details and send back a confirmation.

In summary, this architecture guarantees that various blockchains can collaborate effectively and securely, simplifying the management of transactions across different platforms.

DOI: 10.48175/568

5.2 Relay nodes algorithm implementation

Algorithm 1: Initiated cross-chain transaction.

```
Input Parameters:
(sourceAddress, destinationAddress, sourceBlockchainNum, destinationBlockchainNum,
destinationFunctionName, crossChainData, contractVersion, transactionType)

Check Contract Version
IF version == contractVersion THEN
  Validate Transaction Type
  IF transactionType != 1 THEN
    PRINT "Cross-chain transaction type: transfer"
ELSE
    IF transactionType == 1 THEN
      PRINT "User initiates contract interoperability"
      Create crossChainTxObj using:
crossChainTxObj = CreateCrossChainTx(sourceBlockchainNum, destinationBlockchainNum, sourceAddress,
destinationAddress, destinationFunctionName, crossChainData)
      Store the transaction:
CrossChainTxMap[crossChainTxObj.txNo] = crossChainTxObj
      Trigger event:
EmitEvent("startCrossChainTxEvent", crossChainTxObj.txNo)
    ELSE
      PRINT "Error: Incorrect transaction type"
    END IF
  END IF
END IF
```

The function designed to handle cross-chain transactions takes several inputs, including addresses, blockchain identifiers, a function name, transaction data, the contract version, and the transaction type.

Initially, it verifies whether the provided contract version aligns with the expected version. If the transaction type is neither 0 nor 1, an error message is displayed. If the transaction type is 1, this indicates the user is initiating a contract interoperability procedure. The function then constructs a cross-chain transaction object using the supplied information and appends it to a list of currently active transactions. It also emits an event to signify that the cross-chain transaction has commenced. If the transaction type is 0, it simply categorizes it as a transfer without further actions. This methodology ensures that cross-chain transactions are processed accurately and monitored effectively.

Algorithm 2: Cross-chain confirmation transaction via Relay node.

```
Input: (crossTxNo, txOutcome, contractVer, proof)

IF not isRelayNodeActive() THEN
PRINT("Error: Function restricted to relay node execution.")
ELSE
   IF version.matches(contractVer) THEN
crossChainTxObject = CrossChainTxMapping.get(crossTxNo)
      IF crossChainTxObject.txOutcome == "INITIALIZED" THEN
IF relayNode.belongsToSourceChain() THEN
            IF crossChainTxObject.txType == "TRANSFER" THEN
ParseTransferData(crossChainTxObject)
            ELSE                ParseContractInteroperabilityData(crossChainTxObject)
            END IF
         END IF
      END IF
crossChainTxObject.txOutcome = txOutcome
CrossChainTxMapping[crossTxNo] = crossChainTxObject
crossChainTxObject.proof = proof
EmitEvent("AcknowledgementSent", crossTxNo, crossChainTxObject.txType)
   END IF
END IF
```

A function that enables a relay node to verify a cross-chain transaction. It starts by checking if the function is being called by a relay node; if it isn't, an error is displayed, and the process halts. When the call is authenticated, it confirms the contract version is accurate. The function then fetches the transaction details using the transaction number. If the transaction remains in its original state, it verifies whether the relay node is associated with the source blockchain. Depending on whether the transaction is a straightforward transfer or a contract interaction, it processes the pertinent information. After updating the transaction results and proof, it stores the revised details back in the active transactions list. Ultimately, it sends out a confirmation event to signal that the transaction hasbeen processed. This function plays a crucial role in ensuring that cross-chain transactions are managed appropriately by the relay node.

It guarantees secure and authorized handling by confirming that only relay nodes can validate transactions, thus mitigating the risk of unauthorized access and potential fraud. Furthermore, it upholds consistency by verifying the contract version and only processing legitimate transactions, ensuring that all activities are current and compatible.

Algorithm 3: Inter-node transmission (Track-ability)

```
// Connect to the vehicle tracking system.
ConnectToVehicleTracking(sourcePort, serverNodeAddress, networkHost):

IF serverNodeAddress is not null THEN

networkHost.ConfigureStreamHandler(PID, streamHandler)

// Get the network's listening addresses.
FOR each addr IN networkHost.Network().GetListenAddresses() DO
   port = addr.getProtocolPort(multiaddr.P_TCP)
END FOR
ELSE
// Build a multi-address from the server node's address.
multiAddr = multiaddr.GenerateFromAddress(serverNodeAddress)
vehicleInfo = peer.GetAddrInfoFromMultiAddr(multiAddr)
networkHost.PeerStore().StoreAddresses(vehicleInfo.ID, vehicleInfo.Address, peerstore.ExpireNever)

stream = networkHost.CreateStream(context.Background(), vehicleInfo.ID, PID)

// Establish a non-blocking buffer for reading and writing to the stream.
rwBuffer = buffer.CreateReaderWriter(buffer.CreateReader(stream), buffer.CreateWriter(stream))

Run(writeVehicleData(rwBuffer))
Run(readVehicleUpdates(rwBuffer))

END IF
```

A function for linking vehicles to a tracking server. If the serverNodeAddress is absent, it indicates that a vehicle is initiating a connection. In this scenario, the function establishes a stream handler to control data communication.

It then retrieves the listening addresses for the network to identify available ports. If a specific server address is provided, it forms a multi-address for that server and extracts the vehicle ID. The server's address is subsequently added to a list of known peers for future interactions.

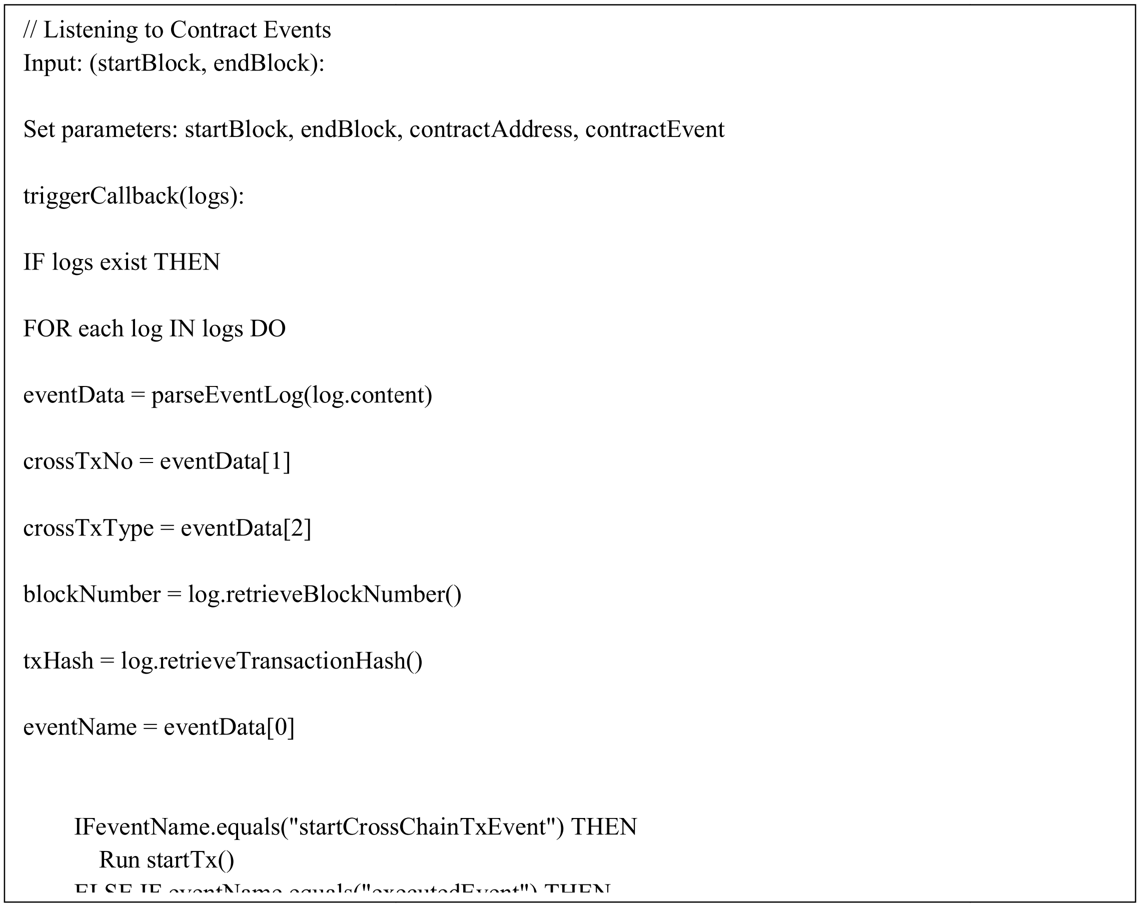
Following this, the function sets up a data stream with the server, creating a non-blocking buffer for both reading and writing data. It initiates threads for sending vehicle tracking data to the server and for receiving updates from the server. This configuration facilitates real-time communication, enabling efficient vehicle tracking and monitoring.

It facilitates real-time communication between vehicles and a tracking server, allowing for immediate updates on vehicle locations and statuses. This is crucial for applications such as fleet management, where timely information is essential for operational efficiency.

This allows for instantaneous updates on vehicle locations and statuses, which is vital for applications like fleet management where timely information is key for operational effectiveness. Additionally, the function fosters a strong peer-to-peer networking model, ensuring that vehicles can connect to servers in a seamless and secure manner. By incorporating server addresses into the vehicle's peer store, it bolsters network reliability and aids in maintaining continuous connections.

The non-blocking read/write streams promote smooth data transfer without interruptions (Uses Hyperledger), enhancing performance and responsiveness. This allows vehicles to persistently send tracking data while simultaneously receiving updates, leading to improved monitoring and decision-making for users.

Algorithm 4: Authentication of relay nodes (Listening to contract events)



```
// Listening to Contract Events
Input: (startBlock, endBlock):

Set parameters: startBlock, endBlock, contractAddress, contractEvent

triggerCallback(logs):

IF logs exist THEN

FOR each log IN logs DO

eventData = parseEventLog(log.content)

crossTxNo = eventData[1]

crossTxType = eventData[2]

blockNumber = log.retrieveBlockNumber()

txHash = log.retrieveTransactionHash()

eventName = eventData[0]


    IFeventName.equals("startCrossChainTxEvent") THEN
      Run startTx()
    ELSE IF eventName.equals("executedEvent") THEN
```

A function to monitor cross-chain events between specified start and end blocks. It begins byestablishing parameters, including the block range and pertinent contract details.

A callback function is created to manage any logs being observed. If logs are available, it goes through each one, decoding the log content to retrieve essential information such as the cross-transaction number, type, block number, transaction hash, and event name.

Depending on the event name, the function executes various actions: if the event is a startCrossChainTxEvent, it triggers the startTx() function; if it's an executedEvent, it calls the execute() function; and if it's a sendAckedEvent, it invokes sendAcked(). This framework allows for adaptable management of different events related to cross-chain transactions, ensuring that the correct actions are taken in response to each observed event.

Validating the relay node is essential as it helps keep the network safe and trustworthy. By ensuring that only legitimate nodes can connect, it prevents unauthorized access and fraud. This guarantees that the data exchanged between blockchains remains secure and reliable, which is crucial for effective cross-chain communication.

VI. CONCLUSION

Existing Intelligent Transportation Systems (ITS) face several significant challenges that limit their effectiveness. Centralized data storage introduces risks, making systems prone to failures, data breaches, and unauthorized access. Scalability becomes an issue as the number of connected vehicles increases, leading to potential bottlenecks, higher maintenance costs, and reduced system performance. These centralized systems are vulnerable to single points of failure, which can disrupt operations across an entire network. Moreover, limited vehicle-to-vehicle (V2V) communication hampers real-time information sharing, impairing situational awareness and responsiveness, especially in congested or accident-prone areas. Inadequate vehicle-to-infrastructure (V2I) communication further limits coordination between traffic lights, sensors, and vehicles, resulting in inefficient traffic flow and delays. Interoperability issues among different ITS platforms and communication protocols complicate seamless integration across regions, hindering large-scale deployment. Poor traceability of vehicle movements and incidents makes traffic management, safety enforcement, and post-incident analysis more challenging, reducing the overall effectiveness of ITS solutions.

To address these limitations, this article proposes an ITS solution utilizing blockchain technology with cross-chain interactions facilitated by relay nodes on a private, permissioned blockchain. Blockchain ensures robust security, transparency, and data integrity, eliminating the risks associated with centralized systems. Decentralized data storage and cross-chain interoperability enable seamless exchange of information between vehicles and infrastructure, enhancing fault tolerance and system resilience. The use of blockchain promotes interoperability among different platforms, encouraging collaboration among public authorities, private fleet operators, and other stakeholders. This framework enhances the efficiency of traffic management, improves real-time decision-making, and ensures reliable, traceable vehicle operations. The proposed solution empowers ITS networks to scale effectively, boosting user trust, adoption, and long-term sustainability by providing a secure, transparent, and integrated transportation ecosystem.

REFERENCES

[1] D. Das, S. Banerjee and U. Biswas ,"A secure vehicle theft detection framework using blockchain and smart contract", Peer Peer Netw. Appl., vol. 14, no. 2, pp. 672-686, Mar. 2023 DOI-https://link.springer.com/article/10.1007/s12083-020-01022-0

[2] Pratik Goswami and Amrit Mukherjee,"AI based energy efficient routing protocol for intelligent transportation system", IEEE Trans. Intelligent Transport System., vol. 23, no. 2, pp. 1670-1679, Feb. 2022. DOI-https://doi.org/10.1109/TITS.2021.3107527

[3] R. Jabbar, M. Kharbeche, K. Al-Khalifa, M. Krichen and K. Barkaoui ,"Blockchain for the internet of vehicles: A decentralized IoT solution for vehicles communication using ethereum", Sensors, vol. 20, no. 14, pp. 3928, 2021. DOI –https://doi.org/10.3390/s20143928

[4] Nisha Malik; Priyadarsi Nanda; Arushi Arora; Xiangjian He ,"Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks", Proc. 17th IEEE Int. Conf. Trust Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE), pp. 674-679, Aug. 2021. Authors - R. Jabbar, M. Kharbeche, K. Al-Khalifa, M. Krichen and K. Barkaoui. DOI-http://dx.doi.org/10.1109/TrustCom/BigDataSE.2018.00099

[5] Balzano W, Lapegna M, Stranieri S, Vitale F. 2022. Competitive-blockchain-based parking system with fairness constraints. Soft Computing 26(9):4151–4162 DOI-https://link.springer.com/article/10.1007/s00500-022-06888-1

[6] Meng B, Wang Y, Zhao C, Wang D, Ma B. 2022. Survey on cross-chain protocols of blockchain.
Journal of Frontiers of Computer Science & Technology 16(10):2177–2192 DOI-http://fcst.ceaj.org/CN/10.3778/j.issn.1673-9418.2203032

[7] Zeng P, Wang X, Li H, Jiang F, Doss R. 2020. A scheme of intelligent traffic light system based on distributed security architecture of blockchain technology. IEEE Access 8:33644–33657 DOI-https://ieeexplore.ieee.org/document/8988221

[8] The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based