SN Computer Science (2024) 5:110
https://doi.org/10.1007/s42979-023-02438-0

ORIGINAL RESEARCH

Integration of Blockchain in VANET Using gRPC for Privacy Preservation of Vehicles

Aditya Kumar Singh[1] · **Jyoti Grover**[1] · **Sumita Mishra**[2]

Received: 19 February 2023 / Accepted: 5 October 2023

Abstract

Vehicular ad hoc network (VANET) is a mobile network comprising vehicles, roadside units, and related infrastructure that enables inter-node communication to manage traffic and enhance road safety. Despite its potential to aid drivers, there are several security and privacy concerns that must be addressed before widespread adoption. It is crucial to validate and hold vehicles accountable in the event of misbehavior while also protecting their privacy and that of their drivers to prevent unlawful tracking and disclosure of personal information. Many current VANET solutions rely on a central trusted authority, which is not a scalable solution and becomes the network's single point of failure. To address these issues, we propose a decentralized blockchain-based authentication solution for VANET that integrates blockchain with VANET using the gRPC framework. This method adds an extra layer of security to the network by ensuring that only authorized nodes are aware of a vehicle's identity. We use blockchain technology to construct a distributed structure and preserve an immutable ledger of data, strengthening the system's integrity. Our technique uses the Hyperledger Fabric, a permissioned blockchain platform, and Veins in OMNeT++ with the gRPC as communication interface. Our proposed approach is more efficient than previous state-of-the-art approaches.

Keywords VANET · Blockchain · Privacy · Security · ITS · gRPC

Introduction

Increased road accidents and traffic congestion are challenges in our day to day lives. According to prominent research scientists, vehicular accidents cause around 85% of mortality and 90% of disability in poor countries each year. Individuals and their families suffer financial damages as a result of road traffic injuries/fatalities [1].

Mobile ad-hoc network (MANET) enables rapid and simple data sharing between mobile devices. The growing notion of the vehicular ad-hoc network (VANET), in which vehicles connect with one another, has boosted the vehicle industry in several ways. VANET is made up of roadside units (RSUs), vehicles' Central Authorities (CAs), and on-board unit (OBUs). The RSUs act as road network interfaces and enable vehicles to send signals across limited range. Each vehicle's OBU is capable of broadcasting the status of the vehicle and gathering data from other vehicles. Dedicated short range communication (DSRC), IEEE 802.11p, wireless access in vehicular environments (WAVE), and Cellular-V2X are the communication protocols utilized in VANET [2]. MAC and physical layers of the DSRC technology are established using the IEEE 802.11p as the default protocol. WAVE discusses the privacy of replaceable data utilizing IEEE 1609 standards.

One of the messages carried over the network with wave service advertisement and wave short message (WSM) is basic safety message (BSM) [3]. Every vehicle in the

This article is part of the topical collection "Research Trends in Communication and Network Technologies" guest edited by Anshul Verma, Pradeepika Verma and Kiran Kumar Pattanaik.

✉ Jyoti Grover
jgrover.cse@mnit.ac.in

Aditya Kumar Singh
2020pcp5598@mnit.ac.in

Sumita Mishra
smishra3@lko.amity.edu

[1] Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur, Rajasthan 302017, India

[2] Department of Electronics and Communication Engineering, Amity University Lucknow, Lucknow, Uttar Pradesh, India

Published online: 18 December 2023

network regularly broadcasts 3–10 BSMs [4] each second which include information like its location, speed, and Vehicle Identification Number. We can also refer to them as periodical messages, but Event-based communications are also delivered by VANET. These messages are only sent in response to specific situations or events, such as congestion, accidents, etc. Excessive traffic, road accidents, and ongoing construction on the road are all common occurrences. VANET is characterized by rapid change in network topology due to high mobility and frequent message sending which is one of the most significant issues with VANET's security [5].

VANETs have numerous security issues that can severely influence the system, causing monetary losses, and can even cost lives. As of now, the authentication framework for VANET is kept up with the use of public key infrastructure (PKI) scheme, where private and public keys are given to vehicles. Every message delivered by a vehicle has its own digital signature and a certificate that is carefully endorsed by the Central Authority (CA), requiring further computation on the reception vehicles' parts to confirm the source of the message. In PKI structure, the delay for decryption and encryption influences the execution of the framework. Consequently, by executing PKI for vehicles validation, the effectiveness of the framework is highly impacted because of more computational overhead required in above approaches. Efficient and Trustworthy VANET requires the following properties [6]:

- Transparency: All participants should be able to see and monitor the network's operations.
- Conditional anonymity: The vehicles identities should be protected and kept private, but authorities should be able to track them down in the event of a conflict or rule breaking.
- Efficacy: The legitimacy of alert messages should be confirmed even if they are sent from a trusted source and even if the network is overloaded.
- Robustness: The system should be able to withstand attacks from outsiders sabotaging the trustworthiness.

Motivation

Vehicular ad hoc networks (VANETs) involve the transmission of various critical messages, which may be periodic or event-driven. To ensure the trustworthiness of both the messages and their senders, the application of blockchain technology can be considered. The decentralized nature of blockchain allows for effective management of confidentiality, integrity, and availability [7]. By utilizing blockchain, vehicles can access historical event information and employ it to create a solution that is not only efficient but also safe.

The most suited method for message authentication and providing a stable network is the public key infrastructure (PKI). The Central Authority (CA) provides a private and public key pair to every vehicle registered in the network under this scheme. Each vehicle uses its private key to digitally sign the message before sending it. The vehicles are also given the public key of the CA and a certificate that contains the public key of each individual vehicle. The certificates are digitally signed by CA using its private key before being sent to the vehicle. Thereafter, vehicles can verify the authenticity of each message using sender and CA's public key.

The computation of two digital signatures can present significant computational challenges for on-board units (OBUs) due to the high overhead associated with this operation. Given the volume of basic safety messages (BSMs) received by vehicles, which can reach up to 10 per second on average, it becomes essential to eliminate such overheads. To overcome this challenge, the proposed solution aims to establish a secure and dependable network that restricts message exchange to only verified vehicles. Achieving this goal mandates the secure storage and management of private identity data belonging to each vehicle. Furthermore, to minimize computational overheads in the system, there is a need to reduce the processing costs for both OBUs and roadside units (RSUs) during digital signature verification.

The peer-to-peer transactions are all stored in a distributed, decentralized system called blockchain [8]. In distributed ledger, each transaction is kept as a block. Immutability, anonymity, transparency, and chronological order are all features of blockchain technology, making it more safe and trust- worthy. Because it adheres to an append-only data type, the data stored in it cannot be changed or erased. Anonymity of vehicle identities are ensured using pseudo IDs for network communication, while private identities of vehicles are maintained in blockchain. When messages are received, RSUs will authenticate the senders using Blockchain, which decreases the vehicle's OBU's processing overhead as well as the authentication delay.

This paper proposes a lightweight solution for authenticating the vehicles using blockchain. We use Hyperledger Fabric v2.4 [9], which is a permissioned blockchain framework to approve the vehicles in VANET. Every participant in a permissioned blockchain is regarded as trustworthy, since they all have current, valid certificates. Through PKI, the members are linked to one another. Every participant maintains or updates a shared distributed ledger for the Pseudo IDs and public key. The messages sent by neighboring vehicles are approved by nearby RSUs. Revocability, access control, validation, and network accessibility are all security features upheld by this framework.

Key Contributions

The following are the major contributions of our article:

- Our paper provides a framework for interfacing hyperledger fabric (a permissioned blockchain platform) and Veins in OMNeT++ with the use of gRPC.
- Blockchain-based decentralized authentication solution for VANET is designed and analyzed using various parameters including the delays caused by adding the Blockchain and Cryptography functions in VANET.
- Comparative analysis of the proposed solution with other state-of-the-art research is also performed.

Organization

The paper is organized as follows: the "Literature review" delves into the literature review, which explores the prior research efforts conducted on the subject of utilizing blockchain technology in VANETs. Section "Background knowledge" describes the required background research for VANET and how the vehicles and its units are identified or authenticated using the Blockchain methods. The proposed authentication architecture and integration of blockchain with VANET simulator is presented in "Proposed architecture". Experimental results are presented in "Experimental setup and simulation results". Finally, "Conclusion" concludes the paper and highlights future work.

Literature Review

The security of VANET using blockchain is presented in [10]. The Inter Planetary File System (IPFS), Ciphertext-based Attribute Encryption (CP-ABE), and the Ethereum blockchain are the foundations for a distributed VANET system suggested in [11]. The blockchain is in charge of managing user identity, and smart contracts are used to keep track of all data. IPFS employs replication proofing to ensure dependability and availability while avoiding individual points of failure. The decryption and encryption procedures are separated in this method by shifting calculation tasks to RSU.

A Blockchain-based unlinkable authentication (BUA) solution was presented by Liu et al. [12], which prevents intruders from compromising the vehicle's security by connecting many messages. The service managers (SMs) are in charge of retrieving vehicle data from the blockchain and confirming the legality of vehicles parked inside their service area in this case. The SMs are also in charge of the vehicle's network registration and conditional traceability. The unlinkability of the communications is increased during the authentication step by randomizing them with a random integer and a timestamp. Network for background review BSMs decrypts the encrypted address and scans the blockchain after getting a message from vehicles. If the address acquired matches the one in the blockchain, SM verifies that the vehicle is valid and sends a signature to it. The vehicle then verifies the SM's identity using the signature it has received, ensuring mutual authentication. The disadvantages of this approach are that it is susceptible to SM adversarial attacks and that the registration process takes longer than authentication.

Lin et al. developed a new blockchain-based conditional privacy preserving authentication (BCPPA) in [13], which addresses difficulties such as private key revocation, frequent contacts, and the need for specialized hardware. VANET uses a key derivation technique in conjunction with an Ethereum-based blockchain to manage certificates successfully. The ECDSA digital signature scheme is employed, which is divided into three phases: system initialization, message verification, and message signing. They have set up an Ethereum test network named Rinkeby to keep track of the transactions.

Reference [14] proposes a decentralized and traceable blockchain-based VANET system to implement a secure authentication method between vehicles and RSUs. This technology offers a safe atmosphere for conversation while preserving user anonymity and preventing genuine identities from being revealed. To avoid the spread of fraudulent messages, the authors created a distributed blockchain-based storage mechanism. They evaluated how well their strategy functioned in terms of accountability, decentralization, compatibility, and several storage options.

Reference [15] presented a blockchain-based decentralized authentication technique using hyperledger fabric which is a permissioned blockchain. They have presented a lightweight authentication system that does not require certificate verification and allows only authorized entities to access the genuine identities of vehicles. Furthermore, when the vehicles get a message, they ask the RSUs to verify or validate the sender, and if the sender is not verified, the message is simply rejected. They also contrasted the latency in authentication between their simulated results and the traditional PKI architecture. This authentication approach eliminates single points of failure and creates a distributed, decentralized system. Congestion in the network is caused by an increase in communication overhead and channel busy time.

The authors [16] presented an anonymous and privacy-preserving authentication mechanism based on MAC. Vehicles acquire a group key during the mutual authentication phase, which they can use to generate and validate authenticated messages. To enable the registration of authentic vehicles and secure communication between them, this group key is created via verifiable secret sharing (VSS).

Reference [17] employs two types of certificates: enrollment certificates issued by multiple pseudonyms and enrollment CA certificates supplied by Pseudonym CA. Pseudonym certificates for the vehicles, which are regarded as short-term certificates, are requested using enrollment certificates. They also have a misbehavior authority, which is in charge of detecting and revoking misbehaving vehicles, reporting misbehavior, and adding them to the certification revocation list (CRL). Table 1 presents related works using blockchain.

Using a permissioned blockchain, such as hyperledger fabric, George et al. suggested a decentralized authentication and identity management method based on blockchain in [15]. They have developed a new way to verify if vehicles are genuine without needing to show proof of ownership or identity. This method is easy to use and can be verified by RSUs.

All of these methods have a number of drawbacks, including increased authentication latency, single-point compromise, increased channel busy time, collusion attack, computational and communication overhead, and channel busy time. We suggest an RSU-based strategy, in which only RSUs are aware of the vehicle's true identity, to get around some of these limitations. To communicate with one another, the vehicles create pseudo IDs once they sign up for the network. The hyperledger fabric (permissioned blockchain) is used to store and manage the identification of each vehicle registered with the network.

Background Knowledge

This section presents summary of VANET and blockchain. It also presents authentication mechanism in VANET and gRPC for interfacing.

VANET

A subclass of MANET is called VANET (vehicular ad-hoc network) in which vehicles are connected with one another. The major goal of VANET is to transmit road and traffic-related information on a regular basis. To avoid a collision or network disorder, vehicles exchange safety information in the form of basic safety messages (BSM). Figure 1 shows a simple architecture and various types of communication which can happen in VANET.

All vehicles participating in VANETs come equipped with on-board units (OBUs), which consist of a diverse set of components, including sensors, wireless communication devices, event recorders, and computational devices [23]. OBUs are responsible for exchanging data with other OBUs and roadside units (RSUs) via IEEE 802.11p [24] radio technology. RSUs are stationary wireless access devices deployed along the roadside, such as traffic lights or streetlights, which serve as intermediary nodes in the network. Furthermore, RSUs are tasked with generating alerts for road hazards, accidents, casualties, crashes, and other

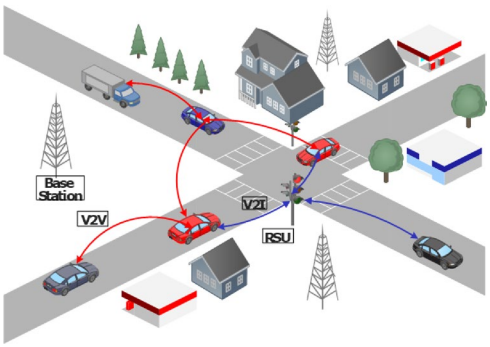


Fig. 1 VANET architecture

Table 1 Related works using blockchain

S. no.	Paper	Proposed method
1.	Leiding et al. [18]	This paper used Ethereum blockchain technology with the proof of stake consensus Mechanism to self manage the network using smart contract
2.	Malik et al. [19]	This paper uses the PKI approach for the authentication of the messages. With the blockchain using the Proof of Authority as Consensus helped in initializing, registering, authentication, and revoking of the vehicles. However, they did not use smart contract for the events
3.	Lasla et al. [20]	This approach uses the bitcoin blockchain in the VANET. It make use of all the entities of the VANET including Centralized Authorities, RSU and vehicles. RSUs are responsible for the authorization and revoking of the vehicles from the VANET
4.	Lu et al. [21]	This approach uses three blockchains for the valid certificates, the messages sent by the vehicles and invalid certificates, respectively. However, the overhead for this approach is very high because of maintenance of these blockchains
5.	Dai et al. [22]	In this approach, the reciprocity of blockchain is used to store the reputation of the vehicles. But lacked in the computation of the overhead related to the calculation and storage of the reputation
6.	Sonia et al. [15]	In this approach, they used the permissioned blockchain maintained by the authentication parties for maintaining the private data of the vehicles, whereas RSU have the read access for the blockchain

safety-critical events via vehicle-to-infrastructure (V2I) communication.

RSUs can connect vehicles to external internet infrastructure, service providers, and trusted third-party entities, thereby enabling active traffic assistance and advanced vehicle tracking. The use of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications facilitates the deployment of diverse applications [25, 26]. Examples of safety–critical applications include collision avoidance, accident alerts, road condition propagation, and weather condition alerts. Additionally, V2I communication supports drivers by providing traffic navigation and information on the availability of parking spaces. Non-safety applications include entertainment features, uninterrupted internet access, and location and map tracking.

In VANET, privacy and security are two major concerns. This is because VANET is a public platform that allows any vehicle to join and broadcast message. Wireless message transmission allows any hostile vehicle to broadcast false information or alter signals transmitted by all other vehicles [27, 28]. These malicious vehicle behaviors may actually put vehicles in danger or disrupt the entire network. Since attackers can track the drivers of the vehicles using the data provided by the vehicles over the network, privacy is an important concern for drivers in VANET. Elliptic curve digital signature algorithm (ECDSA), public key infrastructure (PKI), modified version of TESLA(TESLA++), timed efficient stream loss-tolerant authentication (TESLA), and VANET authentication utilizing signatures are some of the security technologies utilized in VANET. Because of the high speed of the vehicles and the continuously modifying topology of the network, various vulnerabilities are possible in VANET [27, 29–31]. Various types of security vulnerabilities and attacks are discussed in [29, 32].

Authentication in VANET

VANETs should employ security measures, such as authentication, privacy, and security, to enhance trust in V2V and V2I communication. Authentication ensures that only authorized vehicles can access the network and communicate with other network users, with the process being initiated whenever a vehicle joins the network or uses any services, including communication with other vehicles. The following criteria must be satisfied for authentication [29]:

- Overhead in calculation and communication: Vehicle computations, such as cryptographic operations, and the number of queries sent to authenticate a sender vehicle should be kept to a minimum.
- Bandwidth usage: The channel's bandwidth, measured in bytes per second (Bytes/s), must be used to accomplish authentication tasks including the exchange of secret/private keys (SK).
- Vehicle authentication time: It is essential to reduce the amount of time to authenticate vehicles.
- Scalability: The specified authentication system must be able to handle several actions and communications at once.
- Strong authentication: The network should be protected from attacks via authentication methods.

Various authentication mechanisms are utilized in Vehicular Ad-hoc Networks to ensure secure communication, employing different cryptographic algorithms for message signature and verification. The two most commonly used cryptography techniques are public key infrastructure (PKI) and symmetric key scheme (SKS) [33]. The symmetric cryptography system is also known as private key cryptography [29], and it involves the use of a message authentication code (MAC) to authenticate/verify sent messages. Figure 2 depicts a standard representation of a PKI system in VANET.

Employing PKI is considered to be a feasible approach for ensuring security and privacy through authentication in VANETs [34]. As part of this authentication process, vehicles are required to register with a Central Authority (CA). CAs are responsible for issuing certificates, managing vehicle IDs, and distributing public keys to vehicles within their jurisdiction [35]. Each vehicle that joins the network will receive a private key (SK) and public key (PK) pair, as well as a digital certificate, from the CA. The certificate contains the vehicle's public key, a digital signature of the public key using the CA's identity, and the CA's private key.

Each basic safety message (BSM) in VANETs is equipped with the sender's certificate and digital signature, which are generated using the sender's private key. Upon receiving the communication, the recipient will first employ the public key of the CA, which was obtained during the key issuance process, to authenticate the attached certificate. After the certificate is verified, the communication's digital signature

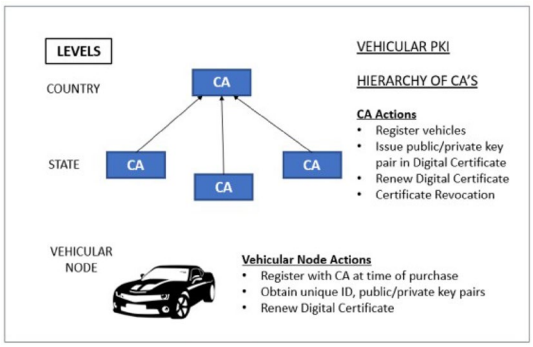


Fig. 2 PKI architecture

is authenticated using the sender's public key. If both the certificate and message signature verifications are successful, the sender of the communication is regarded as authenticated. PKI's primary objectives include ensuring message integrity, authentication, and secure public key distribution [36].

Blockchain

In 2008, Satoshi Nakamoto introduced blockchain technology as a cryptocurrency. This technology features a decentralized, distributed ledger that enables transparent, trustworthy transactions among network users without requiring an intermediary [37]. A blockchain comprises numerous blocks and documents that are cryptographically linked and continuously expanding. It includes network nodes, a distributed database system, a shared ledger, and cryptography.

Each block in a blockchain possesses a unique identifier, a set of transactions, and a cryptographic hash of the current and previous blocks [38, 39]. The hash of each block is obtained by applying a hash function to its contents, and it is challenging to reverse-engineer the original data from the hash. Additionally, the blocks incorporate the current timestamp, which enhances the reliability of the blockchain [40]. As there is no central node on the blockchain to verify the accuracy of the ledgers on the network, the distributed nodes are identical. Hence, consensus algorithms are employed to ensure that the outcomes are precise. With the addition of a new block/record [39], the blockchain is updated. These algorithms enable the network to function in a just and equitable manner.

Blockchain technology is composed of three primary components: cryptographic keys, a shared ledger, and algorithms to store data across a network of blocks. Private and public cryptographic keys are used between two nodes to conduct secure transactions. A shared or distributed ledger helps to store transaction information across multiple nodes in the network, allowing for control by multiple nodes. Mathematical computations are predominantly employed to enhance the security and reliability of the transactions and their storage. Cryptography keys are utilized by blockchain to communicate across the network and verify transactions.

Figure 3 illustrates the functioning of a blockchain, depicting how transactions are generated and uploaded onto the blockchain network. Initially, a node requests a transaction, which is determined by the network's purpose. A block is then created for the requested transaction and broadcasted to all nodes in the network for validation. The transaction is authenticated by designated validators using one of the consensus processes mentioned earlier. If the validators consider the transaction to be valid, it is considered completed after the block is added to the existing blockchain.

gRPC

gRPC, created by Google [41], is a remote procedure call (RPC) framework. It enables us to invoke a server function with minimal payload size. gRPC is beneficial in building distributed applications and services, because a client can directly interact with a server program on a remote machine as if it were a local object.

gRPC is based on creating a service that describes the methods that can be remotely called with their respective arguments and return types. Servers process client requests based on these interfaces, and a gRPC server is started. Clients have a stub, or a client in some languages, that provides the same functions as the server on the client side. Protocol Buffers, which is Google's popular open-source method for serializing structured data, is the default data format for gRPC, but JavaScript Object Notation (JSON) and other formats can also be used. To put it simply, gRPC

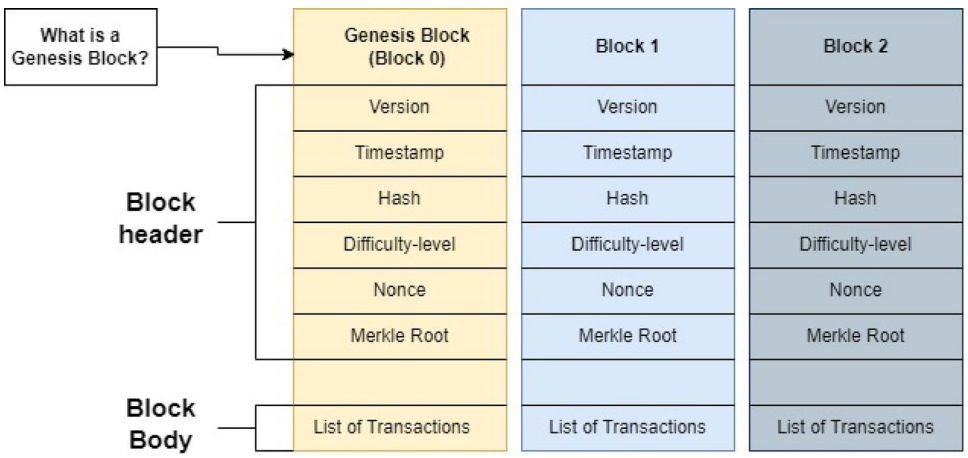


Fig. 3 Blockchain architecture

works by defining a service and the methods that can be called remotely, and clients can call these methods through a stub that communicates with the server.

- One of the initial tasks is to define a message structure that can be utilized as input parameters for the functions. Here is an illustration of the message structure.

```
message block_data{
    string VIN = 1;
    string PsuedoID = 2;
    string Public Key = 3;
    string Status = 4;
}
```

- Once the message structure is in place, the subsequent phase involves defining the server functions that the client will be invoking.

```
service VanetBlockchain{
    rpc reg_vehicle (block_data) returns (response) {}
}
```

- Once the steps have been defined in the proto file, the Proto Compiler is utilized to generate the desired language output. The resulting output from the proto compiler is imported and incorporated by both the client and the server. This enables the population of the data in the structure and the invocation of the service functions that were defined in the proto file.

gRPC utilizes protocol buffers to transmit data between the client and the server by efficiently binarizing the data to reduce the payload size over the channel. The user of gRPC is shielded from the HTTP protocol, unlike in an REST server API, enabling them to focus solely on implementing their application.

Proposed Architecture

The proposed approach utilizes the permissioned blockchain platform, hyperledger fabric [9]. It is a decentralized and distributed framework where RSUs are network participants. The suggested method enables the following operations:

- To obtain a distributed and decentralized framework for storing VANET's private data.
- To provide an uncomplicated authentication approach utilizing digital signatures, public and private keys, and pseudonymous identities.
- To provide straightforward procedures for registering, verifying, and revoking vehicles.

Figure 4 illustrates all the stakeholders involved in the proposed framework. The RSUs play a crucial role in adding blocks to the shared ledger, identifying the vehicle status, and providing the querying vehicle with the Public Key of the authenticated vehicle.

The distribution of the vehicle's registration in the ledger allows all participants to view it, improving network scalability. This means new vehicles can join the network without having to re-register with a different RSU if they travel to a region or domain managed by another RSU.

To register their vehicles with VANET, vehicle owners can visit the nearest RSU and obtain a set of public–private key pairs and a set of pseudo IDs. The pseudo ID is utilized as the sender ID in messages for basic security purposes. The public key, along with the pseudo ID and vehicle status, is stored in the blockchain, allowing RSUs to promptly validate network vehicles. When a VANET vehicle receives information, the RSU verifies the message's pseudo ID to ensure that the sender has a functional public key on the ledger. Additionally, the vehicle maintains a shortlist of valid pseudo IDs and public keys of nearby nodes to reduce transmission costs when relaying messages to the RSU. The list is

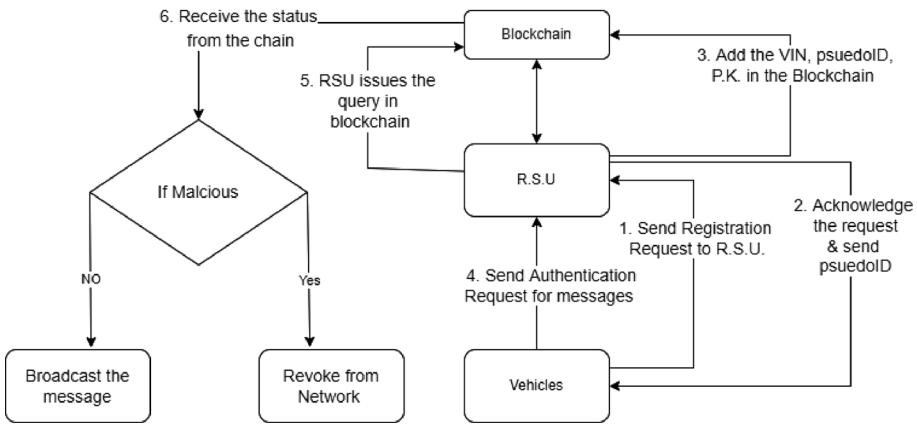


Fig. 4 Proposed architecture

updated with an expiration date for recorded pseudo IDs and public keys to remove outdated entries. This technique relies on certain assumptions, including the following:

- It is assumed that the pseudo-identification numbers of vehicles and the private and public keys issued to road-side units (RSUs) will not be obtained by attackers.
- It is assumed that the registration process will occur in the vicinity of the registering RSU to ensure that registration acknowledgment is not lost.
- It is assumed that each vehicle has an RSU nearby to facilitate message authentication from other vehicles.
- It is assumed that RSUs have ample processing capacity to support vehicle authentication requests and validate digital certificates in received messages to verify message integrity. Furthermore, vehicles themselves have sufficient computational power to perform the necessary computations.

Integration

In the implementation, gRPC [41] was utilized for communication between Veins (written in C++) and the go SDK [42] of the exposed Hyperledger chaincode functions (written in Golang). Figure 5 illustrates this setup, where gRPC utilizes a protobuf schema to transfer messages or data between the two frameworks. The use of gRPC is advantageous, because it is easy to implement with minimal knowledge and significantly reduces overhead during communication. This is achieved by efficiently binary encoding data for transmission across the channel. The steps for integrating Blockchain in VANET are detailed in Fig. 6 and explained in more detail below.

1. Create a Protobuf Schema file (.proto extension).
2. Define functions as services within the schema, such as vehicle registration, updation of details, and querying the ledger.
3. Define messages as arguments for these services, specifying the data structure to be sent to the server and returned to the client (vehicle information in this case).

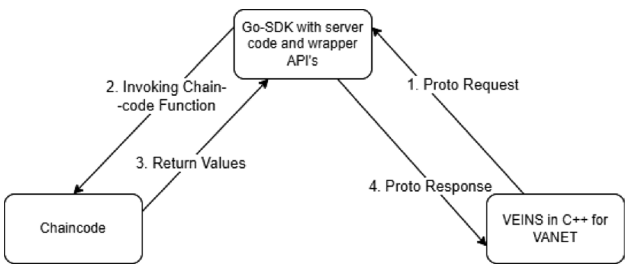


Fig. 5 Integration using gRPC

4. Use the Protoc compiler to compile the schema file for both Golang (Hyperledger go SDK) and C++ (Veins project).
5. Import the compiled files into the relevant codebase to enable function calls from Veins to the Hyperledger go SDK.
6. Call the actual Chaincode functions from the Hyperledger go SDK to execute transactions, such as adding vehicle registration, updating vehicle details, and querying vehicle details.

Chaincode Functions

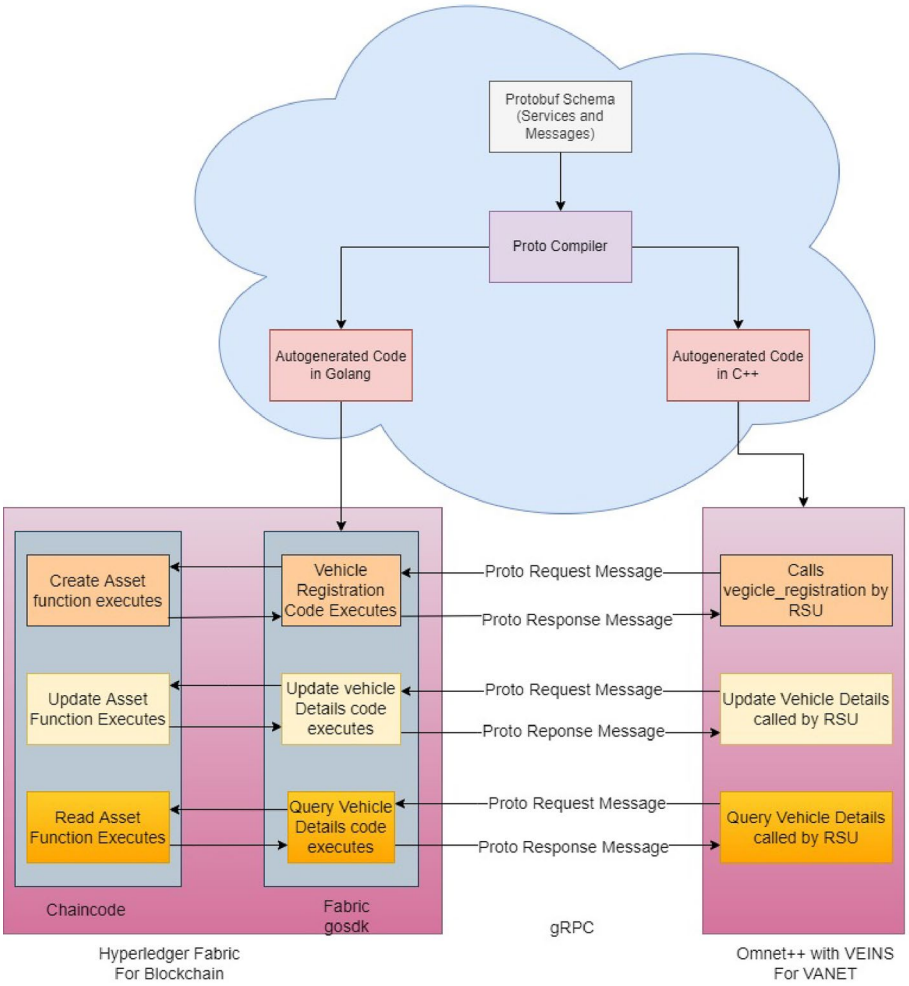
The following functions have been written in the chaincode and can be invoked as needed:

- Vehicle registration: Vehicle owners must register their vehicles with the nearest RSU using a 17-character VIN assigned by the government [43]. RSUs offer unique identifiers, known as PIDs, to vehicles in a VANET for identification purposes, and then securely register them in a blockchain. Furthermore, RSUs generate public and private key pairs for secure communication in the network, with the public key integrated into the PID and the private key safely delivered to the respective vehicle. The vehicle's PID and PK are then updated in the shared ledger, informing all other vehicles and RSUs of its registration.
- Revoke vehicle: If a vehicle behaves maliciously or its message signature cannot be authenticated, the vehicle sends a report to the RSU. The RSU then revokes the vehicle's access to the network and updates its status as malicious on the blockchain.
- Readmit vehicle: After a certain cooling period, a vehicle that has been previously marked as malicious can approach the RSU for re-registration. The RSU will register the vehicle with an updated state in the blockchain.
- Query ledger: When an RSU receives a verification request from a vehicle for a given PID and the vehicle's OBU does not have the details of the vehicle, the RSU invokes a function. The RSU issues a query request to the ledger and receives the block data, which includes the status of the vehicle. The RSU then conveys the details of the vehicle to the requesting vehicle.

Experimental Setup and Simulation Results

In our research, we used several free and open-source software to mimic the road traffic network and capture various characteristics. We employed OMNET++ 5.7 [44], a flexible, modular, component-based C++ network simulator, to simulate the network. We also used SUMO v1.13

Fig. 6 Steps to integrate



[45] to simulate road traffic on extensive road networks. Additionally, we utilized Veins to link the simulation tool OMNET++ with the traffic generator SUMO v5.2 [46] to enable inter-vehicle communication.

To construct the permissioned blockchain architecture, we used hyperledger fabric [47], an open-source platform for creating blockchain networks and applications. The hyperledger fabric blockchain was supported by Hyperledger SDK for GO, Java, and NodeJS, as shown in Fig. 7. Peers invoked chaincode functions to execute transactions, and the Fabric SDK provided a way to execute the chaincode function on behalf of the peer. We wrote the chaincode for the application purpose and modified the SDK provided by the Hyperledger Community to invoke the functions of the SDK directly for executing the chaincode functions. To simulate our proposed model, we leveraged the Crypto++ package [48] to develop the public key infrastructure (PKI) architecture.

In the proposed approach, we have utilized the basic safety messages (BSMs) provided in the veins. Vehicles use them to communicate their status periodically. The

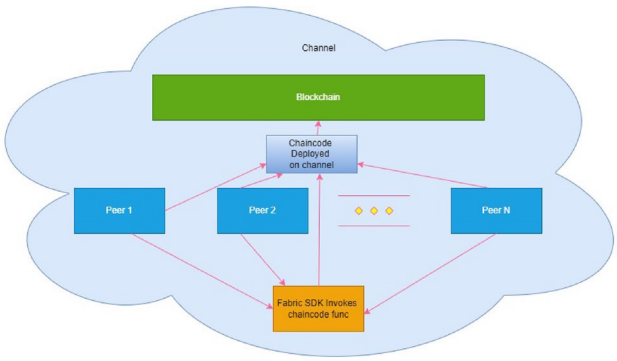


Fig. 7 Hyperledger fabric

computational setup to run the proposed simulation is given in Table 2.

Simulation Parameters

Table 2 Computational setup and software requirements

CPU	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz 2.81 GHz
OS	Ubuntu 20.04.4 (Linux)
No. of cores	2
CPU-Cache	4096 KB
RAM	6144 MB
Software	Omnetpp v5.7, Veins v5.2, Sumo v1.13, Hyperledger Fabric v2.4, gRPC v1.47, Cryptopp v8.6

Table 3 Parameters for simulation

Parameter	Value
Time for simulation	150s
Number of nodes	50
Ground size	2500m
Physical layer protocol	IEEE 802.11p
MAC layer protocol	IEEE 1609.4
Analyzed parameters	Authentication delay, no. of BSM's and RSU request, effect of OBU storage, BSM packet size

Table 3 provides the parameters used for the simulation. The table shows that the simulation time was 150 s, and 50 nodes were included in the simulation. During the simulation, the specified data in Table 3 were captured in a file, and the finish function produced the final values of the simulation parameters.

In our simulation, nodes were added at a rate of one every 3 s. Once a vehicle was created in SUMO, the sumo-launchd.py daemon process would continuously listen for inputs and create the associated OMNeT++ node. The vehicles were generated with a top speed of 50 km/hr, starting from a single location and moving down the road until they reached the end of the route or their top speed limit. If a vehicle encountered traffic, it either slowed down or chose an alternate path. When a vehicle's route was completed, it stopped sending messages, and the finish() function was called to collect data. Additionally, we recorded the values in a text file during the simulation.

The map used in our simulation is depicted in Fig. 8. The simulation outputs were recorded after every 5 nodes were added, at intervals of 3 s, resulting in 10 parts of 15 s each. Upon receiving BSM, our proposed approach mandates that vehicles first check for the PseudoID and public key pair in their OBU storage. If the pair is found, the vehicle will authenticate the signed BSM. Otherwise, it will send a request to an RSU to obtain the PseudoID and public key pair. RSUs with access to the blockchain via gRPC retrieve the details and send them back to the vehicle. The vehicle

Fig. 8 Sumo map

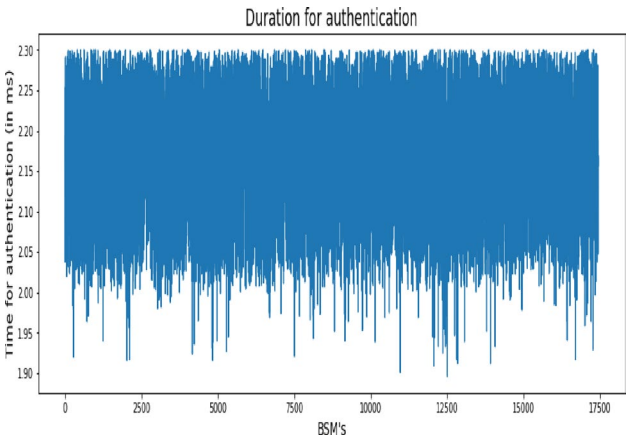


Fig. 9 Authentication delay

authenticates the details and stores them in its OBU storage for a period of time. We varied the storage time of vehicle details and analyzed its effect on RSU requests and authentication delay.

Experimental Results

Our simulation results have been analyzed for multiple parameters. The parameters we have considered include the average time taken to authenticate messages, the size of BSM packets, changes in the number of RSU requests based on changes in the expiry of OBU entries, the message overhead, and the variation in delay with the number of RSU requests.

Authentication Delay

The delay in authenticating basic safety messages was analyzed and the results are shown in Fig. 9. The analysis indicates that the time for message authentication varies from 1.9 to 2.3 ms, with an average delay of 2.15 ms. These

results suggest that the proposed approach offers an efficient means of authenticating messages.

In Fig. 10, we have compared the average authentication time of our proposed approach with the existing ones. George et al.'s Secure identity framework [15] has an authentication delay of 2.0 ms, while Ashghar et al.'s PKI approach [34] has an authentication delay of 3.9 ms. Our approach has almost the same authentication delay as the Secure Identity Framework, but their approach uses a deprecated experimental setup, while our approach uses the latest updated versions of the tools used in the experimental setup.

RSU Requests for the BSMs

At the end of the simulation, we incorporated a finish function which enabled us to obtain the total number of basic safety messages transmitted, as well as the total number of RSU requests received by the RSUs for authentication.

The simulation values were recorded for different numbers of vehicles, ranging from 5 to 50, considering an OBU expiry time of 60 s. In Fig. 11, it is observed that over 50,000 BSMs were transmitted and 2816 RSU requests were sent. The trend shows that the number of BSMs increases exponentially with the increasing number of vehicles, while the RSU requests increase slowly. Therefore, for a large network, it can be concluded that the overhead for RSU requests can be balanced against the security and privacy of the vehicles.

Variation in RSU Request

We have investigated the impact of varying the duration for which vehicles maintain the frequently verified PseudoID and public key pairs on the RSU request transmissions. As described in "Proposed architecture", the proposed approach involves vehicles storing these pairs in their on-board units (OBUs). We recorded the number of RSU requests made by the vehicles while changing the duration of storing the verified pairs in the OBU. The results are presented in the following section.

We conducted simulations for various durations of entry expiry time in the OBU storage. The timestamp of each

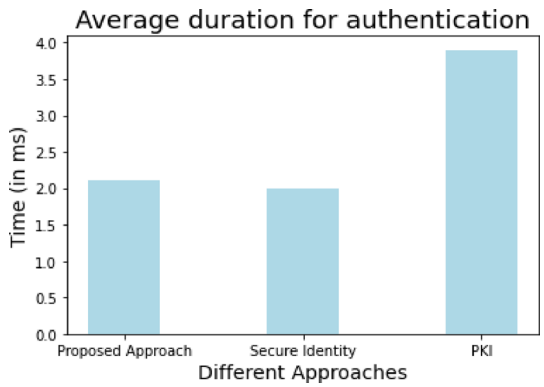


Fig. 10 Authentication delay of different approaches

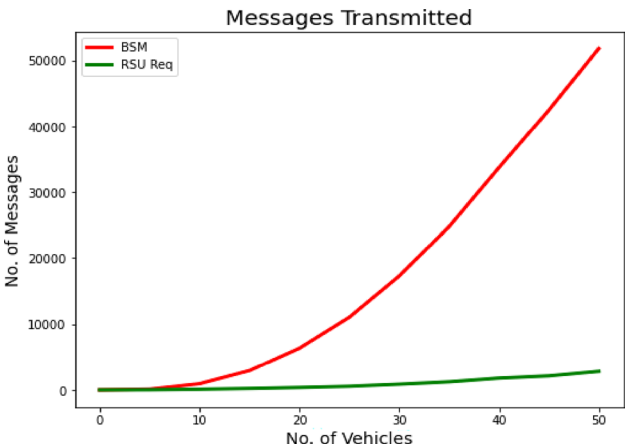


Fig. 11 No. of RSU request and no. of BSM

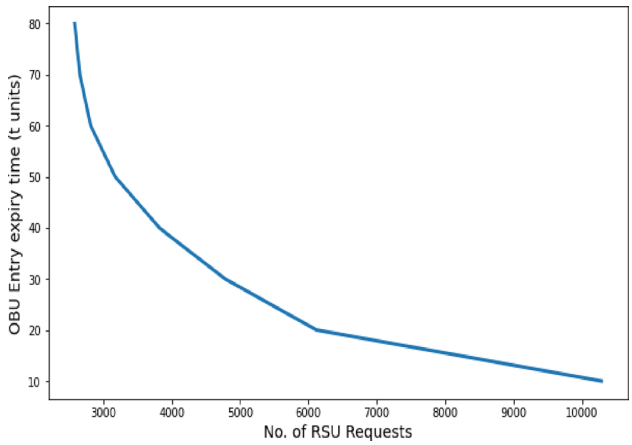


Fig. 12 RSU requests' variation for a scenario of 50 Vehicles

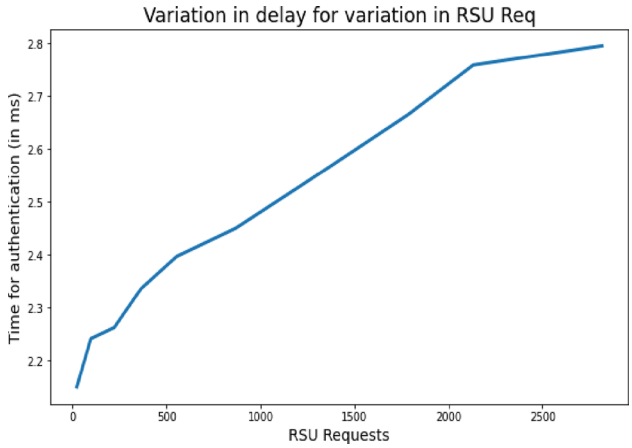


Fig. 13 Effect of RSU request on delay

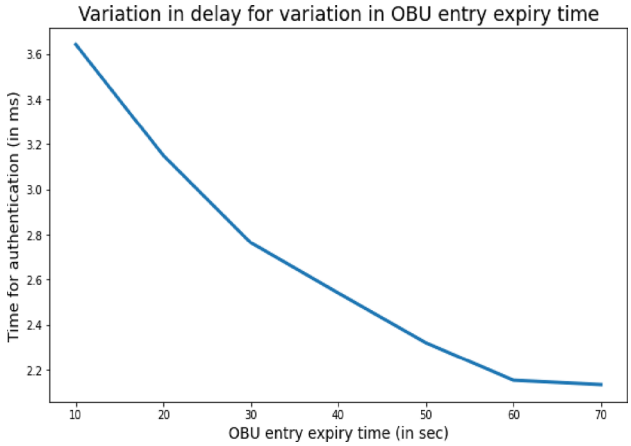


Fig. 14 Effect of OBU entry expiry time on delay

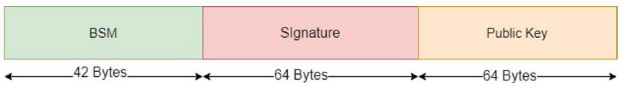


Fig. 15 Basic safety message packet size

added entry was also noted. Upon receiving a BSM, the vehicle checks the OBU storage and removes the expired entries based on the time difference between the current time and the entry time, using t units. We recorded the results for expiry times of $10, 20, 30, \ldots$, and 80 units. The results in Fig. 12 show that decreasing the expiry time reduces the number of entries that the vehicle's OBU can store, resulting in an increase in the number of RSU requests.

Effect of RSU Request on Delay

The impact of RSU requests on the delay in authentication of basic safety messages has also been recorded in our results.

We performed simulations for varying numbers of vehicles, from 5 to 50, and the results were saved in the duration text file using fstream. These values were then used to calculate the average delay caused. As shown in Fig. 13, an increase in the number of RSU requests also results in an increase in delay, as the RSUs need to retrieve the necessary data and transmit it to the vehicles.

Effect of OBU Entry Expiry Time on Delay

For different OBU entry expiry times (10 s, 20 s, 30 s, 40 s,..., 70 s), we ran a simulation for 150 s. The results show that the number of RSU requests is inversely proportional to the OBU entry expiry time. As the OBU entry expiry time decreases, the RSU requests increase, and the time required to authenticate the message is also increased. The relationship between the two is almost linear, and after some time interval, increasing the OBU entry expiry time has no effect on the RSU request for that particular simulation period. This relationship is clearly visible in Fig. 14.

Packet Size of BSM

The proposed approach utilizes a BSM size of 170 bytes, including a signature and a public key of 64 bytes each. However, only specific information is stored in the Blockchain, such as the PseudoID, Public Key, VIN, and the status or misbehavior of the vehicle in the network. The size of the data stored in the Blockchain is illustrated in Fig. 15.

Table 4 provides a comparison between the proposed authentication method, traditional PKI-based infrastructure, and the secure identity framework. It is evident that the average authentication delay is significantly lower compared to the traditional PKI approach and is comparable to the secure identity framework [15]. Additionally, the number of RSU requests, the number of BSMs transmitted, and the channel busy ratio are also considerably lower in the proposed authentication method compared to the other two approaches.

Conclusion

We employed the gRPC framework to effectively integrate blockchain and VANET, providing an additional layer of privacy preservation for message authentication and pseudo identities. The Crypto++ library facilitated the process of key generation for vehicles. RSUs add blocks to the Blockchain, while vehicles maintain a record of frequently contacted vehicles. When there is a miss in vehicle OBU, it queries the ledger through RSU to obtain the details. In contrast to prior techniques that employed the obsolete hyperledger

Table 4 Comparison of proposed authentication scheme with PKI and secure identity framework

	Average authentication delay (s)	No. of RSU requests	No. of BSM's transmitted	Channel busy ratio
Secure identity framework	2.0	7400	80000	0.2
PKI	3.8	9526	100000	0.4
Proposed approach	2.1	2816	50000	0.1

composer, our method distinguishes itself by its utilization of gRPC for integration, a process that is both uncomplicated and efficient. Performance could be further optimized by augmenting the implemented functions. Potential areas of future research might include the identification of misbehavior and the revision of vehicle status on the Blockchain.

Author Contributions All the authors have equally participated in the work.

Funding Not applicable.

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare no competing interests.

Research involving human participants and/or animals This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Organization WH. Global status report on road safety 2018. Geneva: World Health Organization; 2018.
2. Karagiannis G, Altintas O, Ekici E, et al. Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions. IEEE Commun Surv Tutor. 2011;13(4):584–616.
3. Buchenscheit A, Schaub F, Kargl F, et al. A vanet-based emergency vehicle warning system. In: 2009 IEEE vehicular networking conference (VNC). IEEE. 2009. p. 1–8.
4. Cronin B. Vehicle based data and availability. 2022. https://www.its.dot.gov/itspac. Accessed 10 May 2022.
5. Malhi AK, Batra S, Pannu HS. Security of vehicular ad-hoc networks: a comprehensive survey. Comput Secur. 2019;89: 101664.
6. Liang R, Li B, Song X. Blockchain-based privacy preserving trust management model in VANET. In: Advanced data mining and applications. Springer International Publishing; 2020. p. 465–79.
7. Tariq F, Anwar M, Janjua AR, et al. Blockchain in WSNs, VANets, IoTs and healthcare: a survey. In: Artificial intelligence and network applications. WAINA. Springer; 2020. p. 267–79.
8. Li H, Pei L, Liao D, Sun G, et al. Blockchain meets VANET: an architecture for identity and location privacy protection in VANET. Peer-to-Peer Netw Appl. 2019;12(5):1178–93.
9. Foundation H. Introduction to hyperledger fabric. 2022. https://hyperledger-fabric.readthedocs.io/en/release-2.5/blockchain.html. Accessed 10 May 2022.
10. Grover J. Security of vehicular ad hoc networks using blockchain: a comprehensive review. Veh Commun. 2022;34(100):458.
11. Zhang X, Li R, Cui B. A security architecture of VANET based on blockchain and mobile edge computing. In: 2018 1st IEEE international conference on hot information-centric networking (HotICN). IEEE; 2018. p. 2792–2801.
12. Liu J, Li X, Jiang Q, et al. BUA: a blockchain-based unlinkable authentication in VANETs. In: ICC 2020–2020 IEEE international conference on communications (ICC). IEEE; 2020. p. 1–6.
13. Lin C, He D, Huang X, et al. BCPPA: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks. IEEE Trans Intell Transp Syst. 2021;22(12):7408–20.
14. Zheng D, Jing C, Guo R, et al. A traceable blockchain-based access authentication system with privacy preservation in VANETs. IEEE Access. 2019;7:117716–26.
15. George SA, Jaekel A, Saini I. Secure identity management framework for vehicular ad-hoc network using blockchain. In: 2020 IEEE symposium on computers and communications (ISCC). IEEE. 2020. p. 1–6.
16. Lu Z, Wang Q, Qu G, et al. A blockchain-based privacy-preserving authentication scheme for VANETs. IEEE Trans Very Large Scale Integr (VLSI) Syst. 2019;27(12):12–34.
17. Whyte W, Weimerskirch A, Kumar V, et al. A security credential management system for V2V communications. In: 2013 IEEE vehicular networking conference. IEEE; 2013. p. 1–8.
18. Leiding B, Memarmoshrefi P, Hogrefe D. Self-managed and blockchain-based vehicular ad-hoc networks. In: UbiComp '16. ACM; 2016. p. 137–40.
19. Malik N, Nanda P, Arora A, et al. Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks. In: 2018 17th IEEE International conference on trust, security and privacy in computing and communications 12th IEEE international conference on big data science and engineering (TrustCom BigDataSE). IEEE; 2018. p. 674–79.
20. Lasla N, Younis M, Znaidi W, et al. Efficient distributed admission and revocation using blockchain for cooperative ITS. In: 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). IEEE. 2018. p. 1–5.
21. Lu Z, Wang Q, Qu G, et al. BARS: a blockchain-based anonymous reputation system for trust management in VANETs. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications 12th IEEE international conference on big data science and engineering(TrustComBigDataSE). IEEE. 2018. p. 98–103.
22. Dai C, Xiao X, Ding Y, et al. Learning based security for VANET with blockchain. In: 2018 IEEE international conference on communication systems (ICCS), IEEE. 2018. p. 210–15.
23. Mejri MN, Ben-Othman J, Hamdi M. Survey on VANET security challenges and possible cryptographic solutions. Veh Commun. 2014;1(2):53–66.
24. Liang W, Li Z, Zhang H, et al. Vehicular ad hoc networks: architectures, research issues, methodologies, challenges, and trends. Int J Distrib Sensor Netw. 2015;11(8): 745303.
25. Rasheed A, Gillani S, Ajmal S, et al. Vehicular ad hoc network (VANET): a survey, challenges, and applications. In: Vehicular ad-hoc networks for smart cities. Springer; 2017. p 39–51
26. Naja R. A survey of communications for intelligent transportation systems. In: Wireless vehicular networks for car collision avoidance. Springer; 2013. p. 3–35.
27. Phull N, Singh P. A review on security issues in VANETs. In: 2019 6th international conference on computing for sustainable global development (INDIACom). IEEE; 2019. p. 1084–88.
28. Junaid HA, Ali M, Syed AA, Warip M, Nazri M, et al. Classification of security attacks in VANET: a review of requirements and perspectives. MATEC Web Conf. 2018;150(06):038. https://doi.org/10.1051/matecconf/201815006038.
29. Sheikh MS, Liang J, Wang W. A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets). Sensors. 2019;19(16):3589.
30. La VH, Cavalli AR. Security attacks and solutions in vehicular ad hoc networks: a survey. Int J Adhoc Netw Syst (IJANS). 2014;4(2):1–20.

31. Kaur R, Singh TP, Khajuria V. Security issues in vehicular ad-hoc network (vanet). In: 2018 2nd international conference on trends in electronics and informatics (ICOEI). IEEE. 2018. p. 884–89.
32. Tangade SS, Manvi SS. A survey on attacks, security and trust management solutions in vanets. In: 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE; 2013. p. 1–6.
33. Ibrahim S, Hamdy M. A comparison on VANET authentication schemes: public key vs. symmetric key. In: 2015 tenth international conference on computer engineering 'I &' Systems (ICCES). IEEE. 2015. p. 341–45.
34. Asghar M, Doss RRM, Pan L. A scalable and efficient PKI based authentication protocol for VANETs. In: 2018 28th international telecommunication networks and applications conference (ITNAC). IEEE. 2018. p. 1–3.
35. Sakhreliya SC, Pandya NH. Pki-sc: public key infrastructure using symmetric key cryptography for authentication in vanets. In: 2014 IEEE international conference on computational intelligence and computing research. IEEE; 2014. p. 1–6.
36. Zhang Y, Bai X. Comparative analysis of VANET authentication architecture and scheme. In: 2019 12th international symposium on computational intelligence and design (ISCID). IEEE; 2019. p. 89–93.
37. Chatterjee R, Chatterjee R. An overview of the emerging technology: blockchain. In: 2017 3rd international conference on computational intelligence and networks (CINE), IEEE. 2017. p. 126–27.
38. Singh M, Singh A, Kim S. Blockchain: a game changer for securing iot data. In: 2018 IEEE 4th world forum on internet of things (WF-IoT). IEEE; 2018. p. 51–5.
39. Yousuf S, Svetinovic D. Blockchain technology in supply chain management: preliminary study. In: 2019 sixth international conference on internet of things: systems, management and security (IOTSMS). IEEE; 2019. p. 537–8.
40. Narayanan A, Bonneau J, Felten E, et al. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton: Princeton University Press; 2016.
41. gRPC Community. Introduction to grpc. 2022. https://grpc.io/docs/what-is-grpc/core-concepts. Accessed 10 May 2022.
42. GoLang. Hyperledger fabric gosdk. 2022. https://hyperledger-fabric.readthedocs.io/en/release-2.2/fabric-sdks.html. Accessed 8 Mar 2022.
43. Autocheck. What is vehicle identification number. 2022. https://www.autocheck.com/vehiclehistory. Accessed 1 June 2022.
44. Omnetpp. Omnet++ discrete event simulator. 2022. https://omnetpp.org. Accessed 8 March 2022.
45. SUMO. Simulation of urban mobility. 2022. https://www.eclipse.org/sumo. Accessed 8 Mar 2022.
46. Sommer. Vehicles in simulation (veins). 2022.https://veins.car2x.org. Accessed 8 Mar 2022.
47. Foundation H. Hyperledger fabric. 2022. https://www.hyperledger.org/use/fabric. Accessed 1 June 2022.
48. CryptoPP. Crypto++ library 8.6. 2022.https://www.cryptopp.com. Accessed 10 May 2022.