Secure Identity Management Framework for Vehicular Ad-hoc Network using Blockchain

By

Sonia Alice George

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2020

Secure Identity Management Framework for Vehicular Ad-hoc Network using Blockchain

Sonia Alice George

APPROVED BY:

_____

M. Mirhassani

Department of Electrical and Computer Engineering

_____

S. Samet

School of Computer Science

_____

A. Jaekel, Advisor

School of Computer Science

January 15, 2020

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Vehicular Ad Hoc Network (VANET) is a mobile network formed by vehicles, roadside units, and other infrastructures that enable communication between the nodes to improve road safety and traffic control. While this technology promises great benefits to drivers, it has many security concerns that are critical to road safety. It is essential to ensure that only authenticated vehicles transmit data and revoked vehicles do not interfere in this communication. Many current VANET technologies also depend on a central trusted authority that can cost computation and communication overhead and be a single point of failure for the network. By using blockchain technology in VANET, we can take advantage of the decentralized and distributed framework and thereby avoid a single point of trust. Moreover, blockchain technology ensures the immutability of the data strengthening the integrity of the system. In the proposed framework, Hyperledger Fabric, a permissioned blockchain technology, is used for identity management in VANET. All the vehicles with their pseudo IDs are registered, validated, and revoked using the blockchain technology. The vehicles in the network check the validity of the safety messages received from the neighboring nodes, using the services provided by the road side units that have access to the blockchain. This framework works on looking-up the pseudo IDs and public keys on the blockchain for their validity, thus promising a light-weight authentication and reduced computation and communication overhead for vehicles to access the safety messages in the network.

DEDICATION

I would like to dedicate this thesis to my parents, supervisor, internal and external readers who have helped and supported me.

ACKNOWLEDGEMENTS

I would like to thank God for giving me His grace in completing my thesis. I wish to express my gratitude to my parents and my sister for all the support and encouragement they gave me. Further, I wish to thank my supervisor, Dr. Jaekel, and Ikjot Saini for the support and guidance they provided me throughout the research. Finally, I would like to thank Dr. Samet and Dr. Mirhassani for their feedback and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1. Introduction

1.1 Background

As accounted by the Global status report on road safety 2018, released by the World Health Organization (WHO) [1], the annual road traffic accidents have claimed the lives of 1.35 million people. WHO has also recognized road traffic accidents and injuries as the leading killer of people in the 5-29 age group. In 2017, the Canadian Automobile Association (CAA) had identified that drivers in Canada collectively spend over 11.5 million hours and use 22 million liters of fuel per year due to traffic congestion [2]. The technology of Vehicular Ad-Hoc Network (VANET) will be able to mitigate road accidents and traffic congestion.

VANET is a mobile network formed by vehicles, road-side units (RSUs), and other infrastructures. In VANET, vehicles have an on-board unit (OBU) that transmits the state of the vehicles to the other vehicles around it. The RSUs are infrastructures present on the side of the road that help the vehicles to communicate with each other. VANET supports comfort applications and safety applications [3]. Comfort applications provide features like weather information systems, gas/restaurant location, and price details, whereas safety applications support emergency warning systems, lane-changing assistance, and intersection coordination.

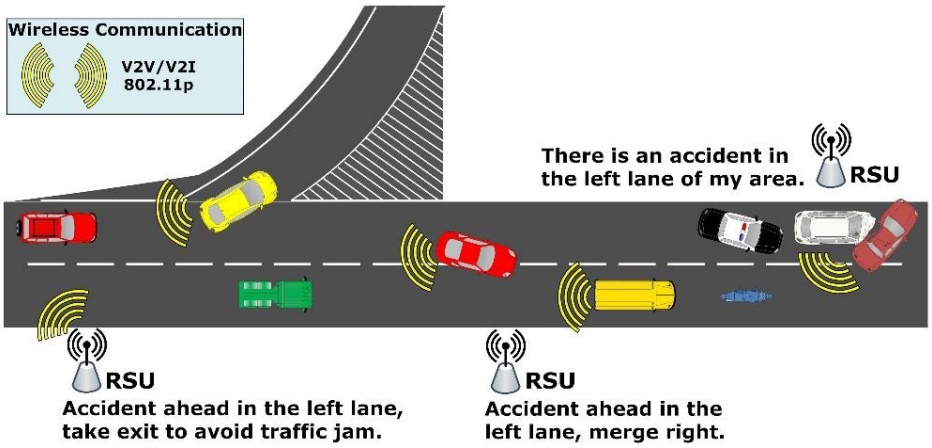

Figure 1.1 Vehicular Ad Hoc Network [4]

Figure 1.1 shows the communication between the vehicles and the RSUs. Communication in VANET can be Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Infrastructure to Infrastructure (I2I), or V2X – vehicle to any other internet-enabled device [5]. In the United States, IEEE has adopted the Dedicated Short Range Communication (DSRC) standard – providing seven 10 MHz channels at 5.9 GHz licensed bandwidth [3]. Vehicular communication is enabled using Wireless Access in Vehicular Environment (WAVE) IEEE 1609 family that provides the necessary protocols and services [6]. IEEE 1609.2 standard provides the methods to ensure the security of WAVE messages and the anonymity and privacy of vehicles.

Security and privacy are a few of the main challenges in VANET. It is essential to ensure that only authenticated vehicles transmit valid messages; otherwise, it can adversely affect the lives of the drives or incur a substantial financial loss. According to [5] attacks in VANET can be:

- Active vs. passive – based on the attacking method; if the attacker performs some malicious action then the attack is active. Otherwise, if the attacker silently listens to the network then it is a passive attack.

- Internal vs. external – based on the membership of the attacker; internal attacks are done by authenticated nodes of the network while external attacks are conducted by nodes that are external to the network.
- Rational vs. malicious – based on the motivation of the attacker; rational attackers try to disrupt the network for their personal benefit, whereas the motive of malicious attackers is to bring harm to the network.

In [5], Abassi et al. has recognized some of the main security requirements in VANET as the following:

- Data integrity – ensuring that the exchanged messages are not altered or modified by an attacker
- Authentication of vehicles – identifying valid vehicles and ensuring that they are who they claim to be
- Privacy and anonymity of vehicles – ensuring that private information of the vehicles is not disclosed to others and not trackable by the attacker
- Vehicle ID traceability – required for non-repudiation to retrieve the identity of the vehicles
- Access control – granting the right access to data and services for different entities in the network
- Revocability – required for identifying misbehaving vehicles and revoking them
- Network availability – ensuring that the network is always accessible
- Network scalability – ability to add other nodes to the network without affecting the performance

Hence, we have used blockchain technology to achieve some of these security requirements, like authentication of vehicles. Blockchain was first introduced by Satoshi Nakamoto in 2008 [7], as Bitcoin cryptocurrency. Blockchain technology supports a decentralized and distributed framework and uses cryptography to store data in an

immutable fashion on a shared ledger. It is deployed on a peer to peer network and uses smart contracts/chain codes to interact with different applications. Some of the earlier blockchain technologies are Bitcoin and Ethereum. These are public/permissionless blockchains where the participants of the network do not trust each other. Now we have private/permissioned blockchains, where additional security is added by controlling the nodes that can form the blockchain network. Using blockchain will have added advantage like providing better traceability as each change in the blockchain can be easily tracked, better network availability as the participants can access the most updated ledger from other peers if its ledger is tampered by an attacker. However, the main focus in our research is to manage the authentication of vehicles in VANET.

1.2 Motivation

Managing the identity of vehicles is a significant challenge in VANET. Further, the current approaches use public key infrastructure (PKI) [8], [9], where each vehicle is assigned a public key and a private key by a Central Authority (CA). These private keys are used to generate digital signatures that will authenticate messages sent by vehicles in the network. Further, each vehicle is provided the public key of the CA and a certificate that contains the public key that is digitally signed by the CA. The receiving vehicles will first verify the digital signature of the certificate with the public key of CA and retrieve the public key of the sending vehicle. Then, this public key will be used to verify the digital signature of the message that is received.

However, these approaches are not efficient in VANET as each vehicle approximately sends a basic safety message (BSM) every 100ms [10] and some of these critical messages will not be authenticated due to the encryption-decryption latency occurred during the narrow time period that they are required the most. As discussed in [8] by Liu et al., authentication based on digital signatures involve more computational overhead for

encryption and decryption processes. From the results in [11], we see that more time is required for processing messages using PKI in VANET.

Hence, we need to have an efficient framework for registering, authenticating, updating, and revoking vehicle IDs. The main objective of my research is to provide a secure and light-weight authentication framework using permissioned blockchain [12] and to manage the identities of the vehicles in the network. Vehicles access the road-side units (RSUs) to validate the pseudo IDs and public keys of other vehicles. The road-side units use the blockchain to have an easy look-up of the pseudo IDs and public keys of valid vehicles. Further, we reduce the computational overhead in the network using a decentralized and distributed framework. Using blockchain will enable us to avoid dependence on a central authority, hence, preventing denial-of-service attacks and a single point of failure.

1.3 Problem Definition

The technology of Vehicular Ad-Hoc Network (VANET) is instrumental in improving road safety and managing traffic control. However, VANETs have many security concerns that can adversely affect the system, incurring financial loss, and costing lives. Currently, the authentication framework for VANET is maintained using public key infrastructure (PKI) where public and private keys are assigned to vehicles. Each message sent by a vehicle has that vehicle's digital signature and a certificate that is digitally signed by the Central Authority (CA), which further requires additional computation by other vehicles for authenticating the sender. Moreover, each vehicle sends a basic safety message every 100ms [10]. The delay in PKI framework for encryption and decryption will affect the performance of the system. Thus, by implementing PKI for authenticating vehicles, the efficiency of the system is affected due to more computational overhead.

Blockchain is a promising future for VANET. It supports a decentralized and distributed framework for the system. A decentralized system will remove the dependence on a central authority to perform important tasks like registering vehicles, revoking vehicle IDs and updating the identities of the vehicles. Moreover, it helps in mitigating single point of failure. Further, having a distributed system will ensure that we have proper backup in case one of the participants is unavailable or if the participant is attacked and loses its data. In this thesis, we aim to provide a secure and computationally efficient authentication mechanism for validating the identities of vehicles sending messages. Here, we have used the PKI framework along with blockchain and used the pseudo IDs and public keys of the vehicles to authenticate them in the network. The pseudo IDs and public keys of the vehicles are stored in the blockchain ledger. The road-side units (RSUs) will have access to the blockchain and can perform a fast look-up to get the pseudo IDs and public keys of valid vehicles. Hence, vehicles can easily authenticate other vehicles by requesting the services from the nearby RSUs.

1.4 Solution Outline

In this thesis, we have proposed a light-weight authentication framework using blockchain for VANET. We have many participants in this network like: authentication parties, road-side units (RSUs) and vehicles. The authentication parties are responsible for registering vehicles and providing them with pseudo IDs, public keys and private keys. They are also responsible for updating the pseudo IDs, public keys and private keys of the vehicles, revoking them if the vehicles misbehave and maintaining the list of valid vehicles in the network. The RSUs are responsible for assisting the vehicles in authenticating the messages sent by other vehicles and reporting any misbehaving vehicles to the authentication parties.

We use Hyperledger Fabric [13], a permissioned blockchain, to maintain and validate the identities of the vehicles. In a permissioned blockchain, each participant joining the

network will be trusted. The participants will be connected to each other through public key infrastructure (PKI). Each participant will maintain a shared ledger that is distributed amongst all the participants. The authentication parties and RSUs have access to the blockchain based on the access control provided to each participant. The RSUs can perform a quick look-up to the ledger to validate the vehicles, and hence, the vehicles will access the nearby RSUs to validate the messages received from other vehicles. This framework supports security features like authentication, access control, revocability, and network availability. Finally, we will simulate this framework and focus on the delay that is caused due to the authentication scheme, and the channel busy time, and compare it to that of the traditional PKI framework. Thus, we can study how the blockchain will affect the performance in VANET.

1.5 Thesis Organization

The rest of the thesis work is organized in the following manner: In chapter 2, we discuss the related work/literature review in managing the identities of the vehicles in Vehicular Ad-Hoc Network using blockchain. In chapter 3, we explain our proposed method to manage the identities of the vehicles using pseudo IDs, public keys and blockchain (Hyperledger Fabric). In Chapter 4, we present the simulation setup and results with its assumptions and analyze the result. Finally, Chapter 5 concludes the research by explaining the insights received during the thesis work and discussing a few of the future works in this field.

Chapter 2

2. Literature Review

2.1 Background on VANET

2.1.1 Introduction

Vehicular Ad Hoc Network (VANET) is a form of mobile ad-hoc network where vehicles are connected to each other. In this type of network, each vehicle has an on-board unit (OBU) that will assist in wireless communication with other vehicles [5]. This network has road side units (RSUs) that are responsible for relaying information over a specified area. However, VANETs are different from mobile ad-hoc networks (MANETs) in the following, as recognized by Yousefi et al. in [3]:

- VANET changes its topology quickly as the nodes are mobile and they commute at high speed.
- VANET has no power or battery restriction unlike MANET where the sensors or other devices have limited power supply.
- VANET can have a large-scale network span especially in cities and highways with more vehicles.

In VANETs, there are mainly four types of communication as below [5]:

- Vehicle to vehicle (V2V) – where one vehicle sends messages to the other vehicles
- Vehicle to infrastructure (V2I) – where a vehicle sends messages to RSU or vice versa
- Infrastructure to infrastructure (I2I) – where infrastructures communicate to one another in the back end to provide services
- Vehicle to X (V2X) – where vehicle communicates to other internet enabled devices

There are mainly 2 types of applications supported in VANET – safety applications and comfort applications [3]. While comfort applications support features like weather information, near-by gas stations or restaurants and other services that improve passenger comfort, safety application mainly focus on enhancing the safety of the passengers. Safety applications include services like providing emergency warning system, lane change assistance and road condition alerts. Many accidents and road side traffic can be avoided using the safety applications in VANET.

Safety applications has two types of messages: event-driven and periodic safety messages [3]. Event-driven messages are generated at the occurrence of an emergency event like an accident or other unsafe situations. Periodic safety messages are beacon messages that are send from a vehicle at fixed intervals. These messages will include the state of the sending vehicle like the position, speed, direction and other parameters. These messages are important for the vehicle to understand its surroundings.

2.1.2 VANET Standards

The standards used in VANET will affect the communication from the physical layer to the application layer as in Open System Interconnection (OSI) model. According to [6],

there are mainly three standards used; Dedicated Short Range Communication (DSRC), Wireless Access in Vehicular Environment (WAVE) and IEEE 802.11p.

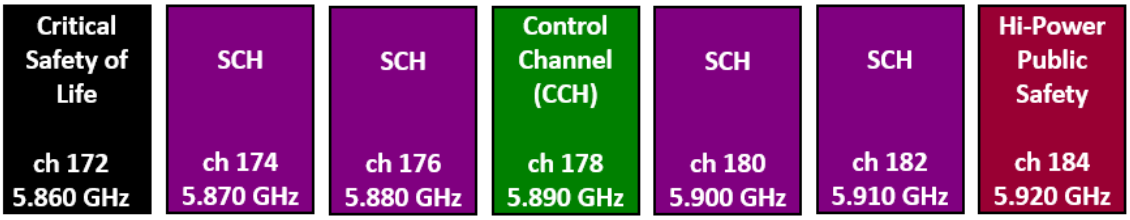

Figure 2.1 DSRC in USA with 7 channels [6]

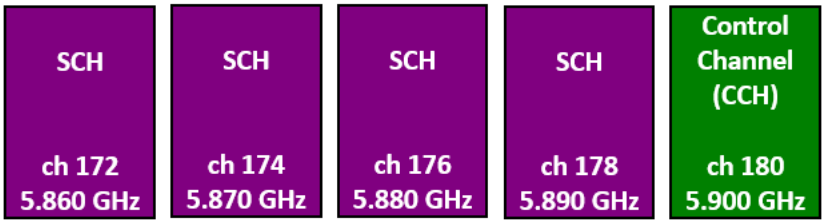

Figure 2.2 DSRC in Europe with 5 channels [6]

In United States, the Federal Communication Commission has dedicated seven 10 MHz channels for Dedicated Short Range Communication (DSRC). This band ranges from 5850 to 5925 GHz as shown in Figure 2.1 [6]. The channel 178 is a Control Channel (CCH) that is used for advertising services on service channels [14]. The other six channels are Service Channels (SCH). The channel 172 is exclusively dedicated for V2V safety communication and accident avoidance and channel 184 is dedicated for high-power, longer-distance communication for public safety applications. However, in Europe, the European Telecommunications Standards Institute (ETSI) has dedicated five channels of 10 MHz with channel 180 for CCH and channels 172, 174, 176 and 178 for SSH. This is shown in Figure 2.2 [6].

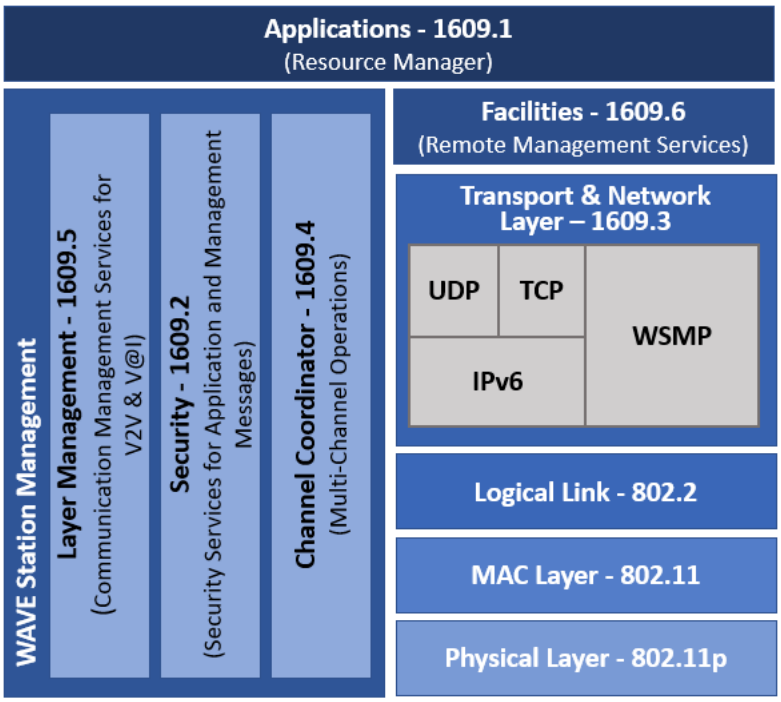

Figure 2.3 WAVE architecture [6]

The Wireless Access in Vehicular Environment (WAVE) IEEE 1609 family specifies the set of protocols, services and interfaces required for intervehicle communication [6]. This architecture will control the Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications. Following are some of the main standards included in the WAVE 1609 family as shown in Figure 2.3 [6]:

- IEEE 1609.1 (Resource Manager): this standard defines the data formats for storage, data flows, resources, and the devices that can be supported by the On-Board Units.
- IEEE 1609.2 (Security Services for Applications and Management Messages): this standard specifies the functions that are required for securing the messages in WAVE and DSRC systems. Further, it supports vehicle anonymity and privacy.
- IEEE 1609.3 (Network Services): defines the services for Network and Transport layers and describes the WAVE Short Message (WSM) protocol.

- IEEE 1609.4 (Multi-Channel Operation): describes the services for multi-channel operation; including Control Channel and Service Channel operations.
- IEEE 1609.5 (Layer Management): this standard is still under development and will aim at managing V2V and V2I communication in WAVE systems.
- IEEE 1609.6 (Remote Management Services): this standard is also under development and aims at supporting the identification of OBU and RSU.

Further, the IEEE 802.11p family provides protocols and services that support intervehicle communication. This standard specifies the definitions for physical and media access control layers for vehicular networks [6].

2.1.3 Security in VANET

Security is one of the main challenges in VANET as it will affect human lives and result in huge financial loss. VANET is prone to attacks as it operates on wireless media for communication. Moreover, the data sent across the network can be used by attackers for tracking the drivers of the vehicles. As mentioned in section 1.1, there are many attacker profiles including active vs passive, internal vs. external, and rational vs. malicious. Further, based on the scope of the attacks, it can be a local attack or a global attack. In local attacks, the affected area is limited, whereas in global attacks, the attack affects many nodes in the network, covering a large area [5].

Some of the main security requirements are mentioned in section 1.1, which include authentication of vehicles, data integrity, privacy and anonymity of vehicles, providing access control, ID traceability, revocability of misbehaving vehicles, network scalability and availability. In [5] Abassi et al. have mentioned other security requirements that are essential for VANET. They are:

- Confidentiality: ensuring that only authenticated nodes can access the messages in the network
- Nonrepudiation: ensuring that the sender of a message cannot deny sending that message

These security requirements are essential to mitigate the attacks that VANETs are susceptible to. In [5] Abassi et al. have recognized four main types of attacks in VANET. They are:

- Confidentiality attacks: these attacks occur when an unauthenticated node has access to the messages being transmitted in the network. Eavesdropping is an example of this type of an attack.
- Integrity attacks: these attacks aim at affecting the validity and usefulness of a message. Examples of this type of attack include message tampering, illusion attacks and timing attacks. In message tampering, the attacker modifies the V2V or V2I messages; in illusion attacks the attacker alters the sensors of the vehicle to transmit erroneous messages in the network and in timing attacks the messages are deliberately delayed thus preventing crucial messages from being transmitted.
- Authentication and privacy attacks: these types of attacks occur when unauthenticated vehicles access messages in the network or when the private information related to a vehicle is tracked [15]. Examples of this type of attacks include sybil attack, impersonation attack, location tracking and identity revealing attacks. In sybil attack, the attacker assumes multiple identities and transmits false messages to legitimate users. In impersonation attacks, the attacker pretends like one of the authenticated vehicles and sends messages. In privacy attacks like location tracking attacks, the attacker builds the vehicle profile by tracking the path of a vehicle and in identity revealing attacks, the attacker gains the owner's or the driver's identity and may get access to their personal data.
- Availability attacks: these types of attack aim at disrupting the services of VANET and making the network unavailable for use. Examples of this type of attack include

denial of service (DoS) attacks, spamming and black hole attacks [16]. In DoS attacks, the entire channel is jammed, and authenticated vehicles cannot access the services of the network. Spamming attacks occur when a spam message is broadcast to the other vehicles increasing the delay of the other transmitted messages and rendering the network unavailable. Black hole attacks occur when the attacker refuses to communicate with other vehicles or communicate using false messages.

In our research, the focus is to manage the authentication aspect of VANET. Here, each vehicle is required to validate the sender of each basic safety message (BSM) to ensure that the sender is authentic and not an attacker.

2.2 VANET Authentication

The current state of art uses the Public Key Infrastructure (PKI) for managing the identities of the vehicles in the network [8] and [17]. Each vehicle in the network will have a public key and a private key assigned to them by a Central Authority (CA). When the vehicles register with the regional authority or the national authority, the OBUs of the vehicles are loaded with sets of public and private keys. Public keys (PK) are visible to all the nodes in the network, while private keys (SK) are only known to the node that it is assigned to and is kept as a secret. Also, during registration, the CA will assign a certificate (Cert) and provide its public key (PK_{CA}) to the vehicles. The certificate will contain the public key (PK_V) of the vehicle V and the digital signature of the public key using the CA's private key (SK_{CA}) as shown in equation 2.1 [17]. The signature also contains the identity of the CA, i.e. ID_{CA}.

$$Cert_V = PK_V \mid Sign_{SK_{CA}}[PK_V \mid ID_{CA}] \qquad 2.1$$

When a basic safety message is being transmitted in the network, the sender of the message computes the digital signature of the message using its private key. Then, the digital signature of the message and the certificate of the vehicle is transmitted along with the basic safety message (BSM). To validate the sender of the message, the receiving vehicle will have to first verify the certificate attached along with the message. This is done using the public key of the CA (PK_{CA}) that was loaded at the time if registration. After verifying the certificate, the sender's public key is obtained and further used to verify the digital signature of the message. If verified successfully, then the sender of the message is authenticated. In [17] Raya et al. represents the communication from a vehicle as below:

$$V \rightarrow M, Sign_{SK_V}[M \mid T], Cert_V \qquad 2.2$$

In equation 2.2, V represents the vehicle sending the message M, at timestamp T. | is used to concatenate the message M with the timestamp T, to ensure that recent messages are being transmitted. $Sign_{SKv}$ represents the digital signature of the vehicle V using its private key, SK_V. $Cert_V$ is the certificate of the vehicle V that was loaded on the OBU of the vehicles during registration. The receivers of the message will have to verify $Cert_V$ and then use the public key obtained to verify the digital signature of the message.

In Figure 2.4, we see that the regional CAs are connected to the national CA. The regional CAs are responsible for registering vehicles by providing public/private key pairs to the vehicles. Further, they are responsible for renewing the digital certificates and revoking them in case a misbehaving vehicle is detected.

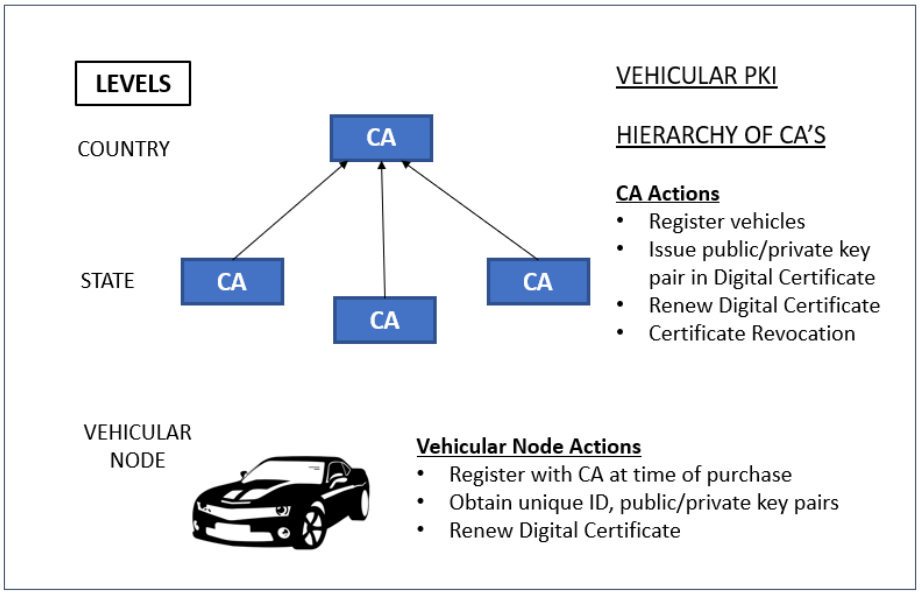

Figure 2.4 PKI Architecture [18]

2.3 Background on Blockchain

The technology of blockchain was first introduced in 2008 by Nakamoto Satoshi as a back-end for peer-to-peer cryptocurrency – Bitcoin [7]. This technology works in a decentralized way, where the data is stored in a shared ledger. This data is cryptographically signed so that the ledger is immutable and can be traced back to the start.

As shown in Figure 2.5, the blockchain technology is similar to that of a physical ledger that will maintain the financial details [19]. Just like the pages of the ledger are connected to each other by page numbers, each block in a blockchain will be cryptographically connected to the previous block. This is done using hash functions. In hash functions, data of any size is taken as input and converted to a hash value of a fixed size. This is a one-way function in that it is infeasible or nearly impossible to retrieve the input data from the hash value. In a blockchain, each block contains the hash of the previous block [19].

Blockchain is different from traditional databases in that it only supports appending information to it and that data cannot be deleted, hence ensuring more transparency.

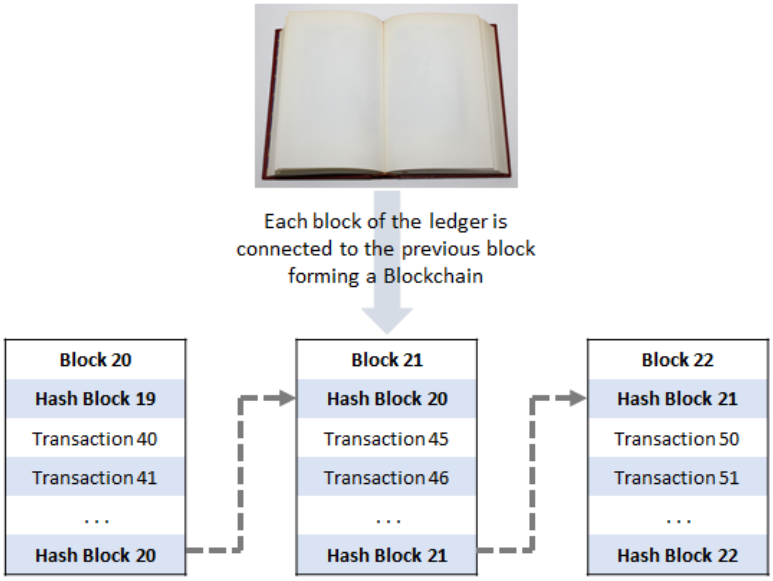

Figure 2.5 Structure of a Blockchain [19]

As shown in Figure 2.6, in a distributed ledger each participant of the network maintains a copy of the ledger. To add a new block to the ledger, all the participants will have to verify the new block, reach a common consensus, and then update their ledger. In blockchains, majority of the nodes need to agree on the new block, to achieve the consensus and to add the block [20]. Some of the popular consensus mechanisms used in blockchains are Proof of Work, Proof of Stake, and Proof of Authority [21], [7].

Thus, if any one of the participants is unavailable due to system failure or DoS attack, the other participants will still be providing services. Once the participant that was down, has resumed its services, it can gain access to the latest ledger by requesting it from one of the other participants [19].

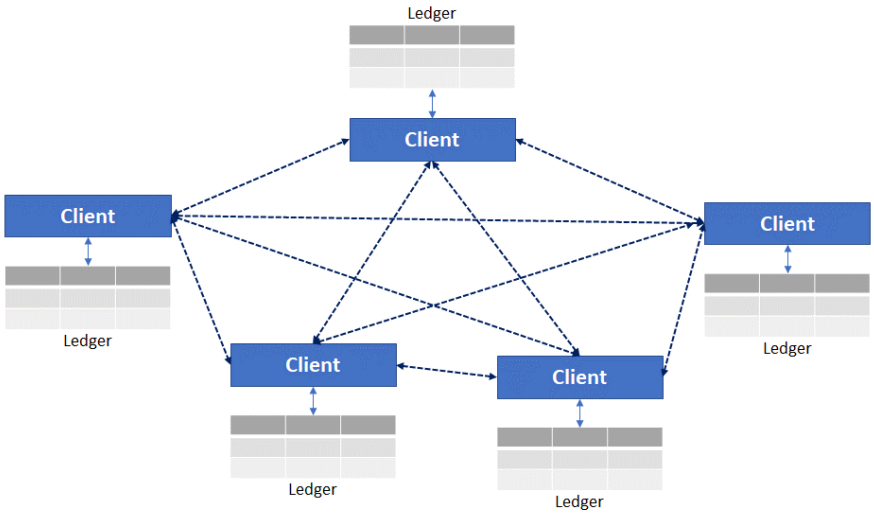

Figure 2.6 Distributed ledger in Blockchain

Currently, there are two types of blockchains: public blockchains like Bitcoins [22], Ethereum [23] and private blockchains like Hyperledger Fabric [13].

2.3.1 Public Blockchain

In public blockchains, any node can join the network and choose to become a miner [12]. Miners are the nodes that validate the new blocks. This is done by competing to solve a cryptographically hard problem. As a reward for mining the new block, the miners will receive an incentive. The nodes in public blockchain can choose to remain in the network or exit out anytime they want. They can also re-enter the network and gain access to the latest ledger. Hence, the data in a public blockchain is available to any node that joins the network [20]. Examples of public blockchains are Bitcoins [22] and Ethereum [23].

2.3.2 Private Blockchain

Currently, various enterprises use permissioned or private blockchains that confirm the identity of the participants of the network. Hence, these types of blockchains do not need to assume that the participants of the network cannot be trusted. In this type of blockchain, only authorized participants with delegated access can join the network [20]. Examples of this type of blockchain will include Hyperledger [13] and Ripple [24].

Hyperledger Fabric is a permissioned blockchain developed by Linux Foundation [13]. It has two components in its ledger: World State – that stores the state of the assets and Transaction Log – that stores the entire log of transactions. It supports Byzantine Fault Tolerance [25] through consensus mechanisms like SOLO, Kafka, or Raft [13]. It supports smart contracts/chain code in different programming languages like Go and Node. Smart contracts are the code that provide controlled ledger access, and they are automatically executed for transactions [26].

2.3.3 Consensus Mechanisms

Blockchains use consensus mechanisms to ensure that the same state of the ledger is maintained across all the participants of the network. This is crucial in a decentralized system. As discussed in [21], some of the consensus mechanisms used in blockchains are:

- Proof of Work (PoW): This consensus algorithm requires a participant node to prove that the work done by that node qualifies them to add the transaction to the blockchain. The node that does the work is called miner. The miners compete to solve a cryptographically hard problem and the first node to solve it receives incentives for the work done. This consensus mechanism requires high computational energy and processing time.

- Proof of Stake (PoS): In this consensus algorithm, the participant node that verifies transaction is selected based on the value of the asset that they place at stake. This mechanism is more energy efficient and fast compared to PoW.
- Proof of Authority (PoA): Here, the validators are provided incentive to maintain their reputation or authority to ensure that they are not compromised. This consensus mechanism is a variation of PoS, where the asset at stake is the authority of the validator [27].

2.4 Using Blockchains in VANET for Authentication

Blockchain is an emerging technology and various use cases of blockchain are being explored. Using blockchain in VANET for authorization is an upcoming research field. Many researchers have used different types of blockchains (public and private) to secure the messages, provide authentication mechanism for vehicles, and to transition the VANET system into a decentralized and distributed framework. Some of these approaches are discussed below.

In [28] Leiding et al. have used Ethereum, a permissionless/public blockchain on VANET. Here, the vehicles, RSUs, and the authorities are participants of the network. This framework uses incentives (ether) to reward the miners. Ether is Ethereum's cryptocurrency. The owners of the vehicles can exchange real world money for ether and use the services. As Ethereum is public blockchain, any new vehicle can join the network easily. However, this approach will use more power for computation as each message sent by the vehicles is stored in the blockchain and will require mining as new blocks are added. Further, Ethereum uses Proof of Stake as consensus where the miner is chosen based on the number of ether, they have put at stake [21].

In [29] Dai et al. has also used a permissionless blockchain, were all the vehicles are participants of this system. Instead of storing the basic safety messages on the blockchain, the vehicles will store their reputation score. This reputation score will be used by other vehicles to authenticate the sender of the message. The reputation score is calculated using indirect reciprocity principle, where a node increases its reputation if it helps other nodes and decreases its reputation if it does not help the other nodes [30]. However, the computational complexity for maintaining the reputation on the blockchain has not been considered to see how useful this system will be in authenticating vehicles. Further, as a greater number of vehicles join this system, it will be hard to reach consensus to add new blocks.

In [31] Lasla et al. has also proposed a blockchain framework for VANET where vehicles, RSUs and authorities form the participants of this framework. It will use PKI to assign public keys and private keys to the vehicles. Here, each message is digitally signed using the sender's private key and the receiving vehicle will authenticate the message, by verifying if the pubic key of the sender is available in blockchain and is indicated as valid. Further, the integrity of the message is confirmed by verifying the digital signature using the sender public key. Moreover, this paper also discusses on operations like revoking the vehicles in case they misbehave in the network. The RSUs will act as the validators and will be responsible for reaching the consensus. However, to verify performance of the framework, Lasla et al. have used Bitcoin system. Bitcoin framework use Proof of Work consensus, which is slow due to the high computational requirement to add new blocks [21].

In [7] and [32], permissioned blockchains have been used with VANETs. In [7] Malik et al. have used PKI to generate public and private keys for the vehicles. Using these keys, the vehicles will authenticate themselves with the RSUs through encryption and decryption. The pseudo ID of the vehicle is digitally signed by the CA and added as blocks in the blockchain. However, these blockchains do not use smart contracts. Smart contracts

are programs that automatically run on the blockchain and are used to ensure that all the participants follow the agreement of blockchain usage [26]. Further, in [7] Proof of Authority is used where the validator's identity is kept at stake [33]. In [32] Lu et al. has proposed a framework using 3 different blockchains with VANET; for storing the valid certificates, revoked certificates and the messages transmitted by the vehicles. For authenticating vehicles, their certificates are searched on the blockchain that stores valid certificates and further searched on the blockchain that stores the revoked certificates. The vehicle will be authenticated if its certificate is present in the former blockchain and absent in the latter one. This shows that the vehicle has not been revoked after the certificate was issued to it. Further, RSUs will act as validators of this framework and they use the Proof of Work consensus, which requires high computation and more processing time [21]. Table 2.1 shows a brief description of each of the papers discussed above.

Table 2.1 Summary of related works

No.	**Paper**	**Proposed Method**
1.	Leiding et al. [28]	To deploy VANET using Ethereum blockchain technology using Proof of Stake as consensus. The proposed system is incentive based.
2.	Dai et al. [29]	To use Indirect Reciprocity to store the reputation of vehicles in the blockchain. But have not considered the computational and storage overhead for processing the reputation.
3.	Lasla et al. [31]	The blockchain framework has all entities like authorities, RSUs and vehicles connected to the blockchain and RSUs act as the validators for this network. The RSU lookup for authorization of vehicles was tested using Bitcoin blockchain.
4.	Malik et al. [7]	Proposed a blockchain framework for VANET using PKI. The blockchain uses Proof of Authority as consensus and provide operations for initializing, registering, authenticating and revoking vehicles. However, this blockchain does not use smart contract or events.

No.	Paper	Proposed Method
5.	Lu et al. [32]	The proposed framework has 3 different blockchains – for valid certificates, revoked certificates, and messages send by the vehicles. Law Enforcement Authority keeps the Vehicle Identification Number (VIN) and public/private keys relation and it is kept confidential to provide anonymous authentication. RSUs act as the validators of the network and use Proof of Work consensus.

Moreover, Table 2.2 shows how this research is different from the existing approaches.

Table 2.2 Comparison of different proposed methods

Property	Blockchain Type	Blockchain Participants	Consensus Used	Simulation Implemented	Other
Leiding et al. [28]	Ethereum - Permissionless	Authorities, RSU, vehicles	Proof of Stake	✗	Incentive based framework – using Ether
Dai et al. [29]	Permissionless	Vehicles	Not mentioned	✓	Indirect reciprocity for reputation
Lasla et al. [31]	Bitcoin - Permissionless	Authorities, RSU, vehicles	Proof of Work	✓	Using PKI – public and private keys
Malik et al. [7]	Permissioned	Authorities, RSU	Proof of Authority	✓	Using PKI – public and private keys
Lu et al. [32]	Permissioned	Authorities (Law), RSU, vehicles	Proof of Work	✓	3 blockchains, reputation evaluation

Property	Blockchain Type	Blockchain Participants	Consensus Used	Simulation Implemented	Other
Proposed method in this research	Hyperledger Fabric - Permissioned	Authorities, RSU	SOLO, Kafka or Raft	✔	RSUs can be queried to validate the vehicles

Apart from these proposed methods, there are additional literatures that have discussed the use of blockchains in VANET. In [34], Lei et al., have discussed how blockchains can be efficient than traditional PKI architecture with a trusted third party, specially when the vehicles commute from one security domain to another. A security domain is the area handled by security managers, who are responsible for managing the cryptographic keys for the vehicles. When vehicles pass from one security domain to another, there are additional operations required to authenticate the vehicle. The CAs at different security domains will have to exchange the cryptographic keys of the vehicle and perform several handshakes, making the key exchange inefficient. However, using blockchain can eliminate this requirement, as the keys can be stored on the shared ledger and accessed by all the participants.

In [35], Jiang et al., proposed a framework with blockchain in VANET, where five blockchains are used based on the data stored on the blockchain. These different blockchains also have different nodes as participants. Here, the Blockchain 1 is maintained by the RSUs and their neighbors, Blockchain 2 is maintained by vehicles and RSUs, Blockchain 3 is generated by RSU and the neighboring RSUs, Blockchain 4 is maintained by RSUs and toll station nodes, and Blockchain 5 is maintained by vehicles, gas stations and charging stations. However, as this framework has vehicles as a node, the transaction rates and the time to reach consensus will be higher.

In [36], Wang et al., have proposed a framework with two blockchains; one maintained by the Registration Authorities and the other maintained by the Certificate Authorities. The Registration Authority is responsible for encrypting vehicle identities and storing them, while the Certificate Authority is responsible for generating the certificates related to the authentication details and other digital certificates. However, this framework was not tested or simulated for checking the performance of the system. In [37] Decoster et al., proposed a blockchain based framework in VANET making the system forensic ready while maintaining the privacy of the users. Here, vehicles are a part of the blockchain and will store the hash of the messages along with the digital signature of the sending vehicle. This will enable the ledger to be forensic ready, as the node that generates the message will have its digital signature in the blockchain along with the hash of the message.

In this thesis, we have used permissioned blockchain – Hyperledger Fabric [13] which is an emerging blockchain technology. Here, the participants of the blockchain system are RSUs and authorities. We have used pseudo IDs along with PKI public and private keys. The RSUs will be assisting the vehicles to authenticate other vehicles by providing a quick look-up for the pseudo IDs assigned to the vehicles and checking if the public key assigned to the vehicles are valid. Further, as discussed previously in section 2.3.2, Hyperledger Fabric has a plug-in to incorporate different consensus mechanisms like SOLO, Kafka or Raft.

Chapter 3

3. Proposed Method

3.1 Introduction

Vehicular Ad-Hoc Networks (VANETs) have great potential in improving traffic control and mitigating road accidents. This is done by sending messages or basic safety messages (BSMs) between vehicles (V2V communication) or between vehicles and road side units (RSUs) (V2I communication).

Security is one of the main challenges in VANET. This is crucial because it directly affects the lives of the commuters or incurs a substantial financial loss to them. As discussed in section 2.1.3, there are many security concerns and different potential attacks that could take place in VANET. Authentication is one of these security requirements that is very crucial for validating the messages from a sending vehicle. It is important to improve the efficiency of the authentication scheme so that a higher number of messages can be validated by the vehicles, which will, in turn, ensure that critical safety messages are not dropped due to the delay in the authentication.

The traditional public key infrastructure (PKI) architecture used with VANET includes encryption and decryption to generate and validate the digital certificates attached along with BSMs. This will result in more delay to authenticate the node sending messages. As

discussed in section 2.2, below are the main steps included in authenticating the messages sent by other vehicles:

- On receiving each BSM, the receiving vehicle uses the public key of the CA (PK_{CA}) stored in its on-board unit (OBU) during registration, to validate the certificate of the sending vehicle, $Cert_V$. The format of the $Cert_V$ is given in equation 2.1. It contains the public key of the sending vehicle, digitally signed using the private key of the CA, SK_{CA}. Hence, the public key of the CA, PK_{CA} is used to verify it.
- After verifying the certificate of the sending vehicle, its public key (PK_V) is used to verify the digital signature attached along with the BSM. The format of a BSM is represented by equation 2.2. To verify the digital signature that was generated using the private key of the sender, the public key of the sender is used.

Both above-mentioned processes that validate the digital signatures require considerable computation for verifying them using the respective public keys. Hence, in our proposed framework, we have implemented a light-weight authentication scheme for VANET. Instead of using the traditional PKI, which uses public keys, private keys, and certificates for the nodes, we have used this along with a permissioned blockchain. We have proposed the use of pseudo IDs along with public keys to provide a quick look-up for authentication from RSUs. Further, this framework is implemented on Hyperledger Fabric, implementing VANET in a decentralized and distributed fashion. Blockchains provide an immutable record of information and an append-only database, ensuring traceability and transparency of data.

3.2 Synopsis of Authentication using Blockchain

Many authors have discussed the use of blockchains in VANET for validating messages and maintaining the registration and revocation of vehicles [31], [28], [32], [7], and [29]. Some of these approaches use public blockchains like Bitcoin [31] and Ethereum [28].

They use consensus mechanisms like Proof of Work, which is a relatively slow consensus algorithm [21]. Hence, there will be a delay in updating data on the blockchain for all the ledgers and will lead to a slower system. This will adversely affect the efficiency of basic safety messages being authenticated and received by other vehicles on the network.

Some of the papers mentioned in section 2.4, have used PKI architecture. In this framework, public and private keys are used for encryption and decryption functions, and to generate and verify digital signatures. However, this leads to higher processing time and delays due to the computation time required for encryption and decryption. Further, in our research we use blockchain technology with the public and private keys. Here, the pseudo IDs and the public keys of the vehicles are stored in the shared ledger across all the participants (RSUs and authorities). The architectures in [29], [31], [32], and [28], have each vehicle in the network as participants of the blockchain. While this provides shared ledger access to the vehicles, it also reduces the performance of the blockchain due to more participants in the network. As the number of participants increase, the processing time to reach consensus also increases. Further, storing the shared ledger on each vehicle in the blockchain network will increase the storage overhead in vehicles. Thus, in our research we have the authorities and RSUs as participants of our blockchain, and not the vehicles.

Further, in [7], one of the parameters used for evaluating the performance of the proposed method is the delay in the RSU communication measured in milliseconds (ms). This parameter is also our primary focus for measuring the performance of the proposed light-weight authentication framework using Hyperledger Fabric and pseudo IDs. We aim to compare our results to the traditional PKI architecture using Elliptic Curve Digital Signature Algorithm (ECDSA). We have used ECDSA as it is widely used across VANET due to the high security provided in less key size [38].

3.3 Architecture of Proposed Method

In our proposed architecture, we use Hyperledger Fabric, a permissioned blockchain, developed under the Hyperledger project. As discussed in section 2.3.2, Hyperledger Fabric implements a decentralized and distributed framework. The participant of the blockchain network are the Authentication Parties and the RSUs. Through this proposed method, we aim to accomplish the following:

- Provide a decentralized and distributed framework for VANET
- Provide a light-weight authentication scheme using pseudo IDs, public/private keys and digital signatures
- Provide transactions to register, validate, and revoke vehicles

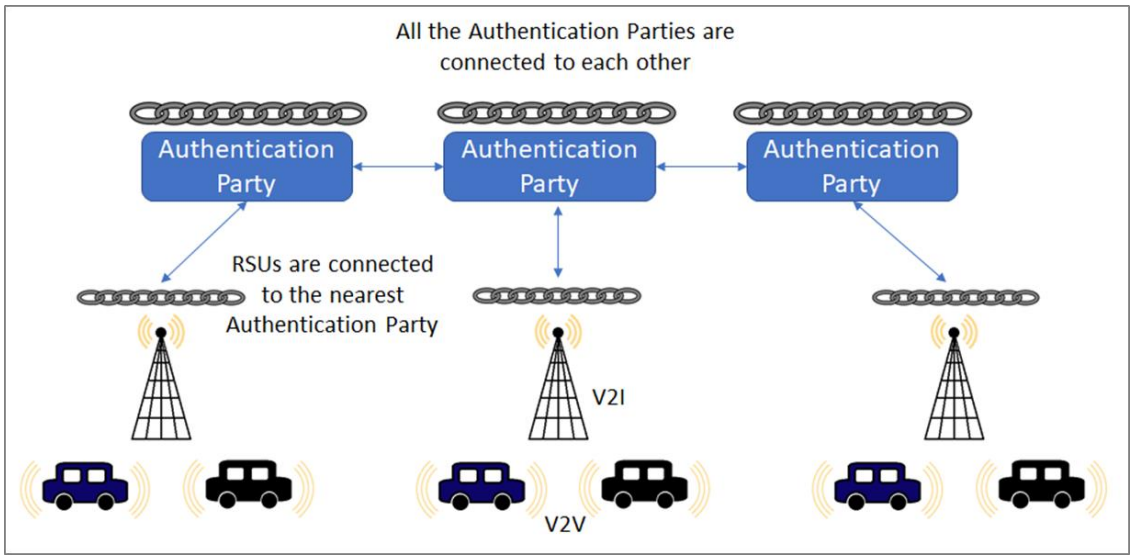

Figure 3.1 Proposed architecture

Figure 3.1 shows the participants in the proposed framework. The Authentication Parties and the RSUs have access to the blockchain. However, the RSUs are provided read-only

access to the shared ledger, while the Authentication Parties are given full access so that they can submit transactions to update the ledger.

In this type of decentralized network, we have more than one Authentication Parties, so that owners of the vehicles in different regions can register with their nearest Authentication Party. Further, as the ledger is distributed, the vehicle's registration in the network is visible to all the participants (other authorities and RSUs) and not just the authority registering the vehicle. This accounts for better network scalability as new vehicles can easily join the network and do not need to register again with another authority in case they commute to a region or domain maintained by another Authentication Party.

The owners of the vehicles register them in VANET by going to the nearest Authentication Party. As a part of registration, each vehicle receives a set of pseudo IDs along with a set of public-private key pairs using ECDSA. The pseudo ID is used as the sender ID in basic safety messages. The public key along with the pseudo ID of the vehicles are stored in the blockchain to provide a quick look-up for the RSUs to validate the vehicles in the network. Each time a vehicle receives a message in VANET, the pseudo ID in the message is verified with the RSU to check if the sender has a valid public key in the blockchain. The vehicle also maintains a short list of valid pseudo IDs and public keys of its recent neighboring nodes, to reduce the communication overhead in transmitting messages to the RSU. This list is maintained along with an expiration time for the recorded pseudo IDs and public keys to ensure that old entries are deleted.

3.3.1 Assumptions

Our proposed method has few assumptions. They are listed below:

- We assume that the Authentication Parties issuing the cryptographic keys like the public and private key pairs, and the pseudo IDs for the vehicles, and the road side units (RSUs) will not be compromised by the attacker.
- We also assume that there is an RSU within the range of each vehicle to assist with the authentication of other vehicle's messages.
- We assume that the RSUs have enough computational power to assist the vehicles requesting to validate other vehicles and for verifying the message integrity by validating the digital signature in the messages received by the RSU. Further, the vehicles too have enough power to perform the necessary computations.

3.3.2 Operations at different nodes

There are mainly three different types of nodes in our proposed model. They are the Authentication Parties, RSUs and the vehicles. Below are the functions taking place at these nodes.

Authentication Party:

- The owners of the vehicles register them with the nearest Authentication Party by providing the vehicle's unique 17-character Vehicle Identification Number (VIN) [39].
- The Authentication Party then generates a set of public-private key pairs and pseudo IDs (PK_V, SK_V and PID_V) for each vehicle.
- Then this information is submitted as a Registration transaction to the blockchain, which is further discussed in section 3.3.3 below. Once the transaction is submitted to the blockchain, the data is available to other participants of the blockchain; i.e. RSUs and other Authentication Parties.
- We have used the default consensus of SOLO, where there is one ordering node that will order the blocks in the blockchain and update all the ledgers in the

blockchain. However, this is not fault-tolerant and is only used for simulation purposes. More research is required to choose a fast and fault-tolerant algorithm for reaching consensus.

- The RSUs will contact the Authentication Party to report a vehicle that is compromised, and the Authentication Party will execute the Misbehaviour transaction as discussed in section 3.3.3 below.

RSU:

- The RSUs will have access to the shared ledger and hence, can access the most updated information on vehicles. However, they have read-only access to the ledger and cannot send transactions to the blockchain.
- When the RSUs receive a BSM, it checks if the PID_V and the PK_V of the vehicle V is valid in the blockchain and if valid then, the signature of the message is validated, by retrieving the corresponding public key from the ledger.
- However, if the PID_V or the PK_V are invalid in the blockchain or if the message signature is invalid then, the RSU suspects that the vehicle identity might be compromised and then report it to the Authentication Party.
- The vehicles in the network will also request services from the RSU to authenticate the sender of the message. The RSUs will receive the PID_V and PK_V of the sender from the requesting vehicle and will check if that PID_V and PK_V are present in the blockchain with valid status. The result of the validation is then broadcasted to all the vehicles.

Vehicle:

- During the time of registration, each vehicle receives a set of pseudo IDs, public/private key pairs (PID_V, PK_V and SK_V respectively) from the Authentication Party.

- When a vehicle sends a BSM, it sends additional information like its pseudo ID (PID_V), public key (PK_V), and the digital signature of the message generated using the private key of the vehicle (SK_V). This is represented by equation 3.1, where M is the basic safety message (BSM), T is the timestamp, $Sign_{SKv}(M)$ is the digital signature generated using the private key, SK_V.

$$V \rightarrow M, Sign_{SK_V}(M|T), PID_V, PK_V \quad 3.1$$

- When a BSM is received, the vehicle will verify the PID_V and PK_V of the sender in the list of valid PIDs and PKs maintained locally. If the PID_V and PK_V are recognized as valid and the signature is verified, then the sender is authenticated. If PID_V and PK_V are unknown, the vehicle will send a request to the RSU nearby to verify the PID_V and PK_V, and if validated then it is added to the vehicle's local list of valid PID_V and PK_V along with an expiration time.
- When the vehicles query the RSU to validate the PID_V and PK_V of a vehicle, it waits till the channel is free and listens if other vehicles query the same PID_V and PK_V to the RSU and receives a reply. If no other vehicle query for the same PID_V then, the vehicle requests the RSU to validate it.

3.3.3 Operations using Blockchain

The primary operations done using the blockchain are the following:

- Registration – the owners of the vehicles are required to register them with the nearest authority. The authority records the Vehicle Identification Number (VIN) of the vehicle, which is a 17-character unique identification for each vehicle [39]. Further, the authority issues a set of pseudo IDs (PIDs) for the vehicle that are used as the identity of the vehicle in VANET. The Authentication Party maintains a

mapping between the VIN and the set of PIDs assigned to each vehicle. Further, a set of public-private key pairs (PK/SK) are also generated using ECDSA and communicated in secure manner to the vehicle. The shared ledger is then updated with the vehicle's PID and PK, and all the other authorities and RSU know that the vehicle is registered in the system.

- Misbehavior Report – when the RSUs detect a compromised vehicle or if there is a suspicious vehicle in the network, then the RSUs send this vehicle's pseudo ID to the nearest Authentication Party, and the authority submits this transaction.
- Revocation – if the misbehavior report exceeds a certain threshold, for example, three, then this transaction is automatically called to revoke the vehicle from the network. The data related to the vehicle is not deleted from the blockchain but updated to reflect that the vehicle is invalid in the network.
- Readmission – When a revoked vehicle's owner approaches the authority to register back to the system, this transaction is called to readmit the vehicle to the network. The vehicle will be provided with a set of pseudo IDs (PIDs) and public/private keys (PK/SK). The PIDs and the PKs are updated on the blockchain to indicate as valid in the network.
- Query – this is used by the RSUs to check if a vehicle is valid in the network based on the PID and PK of the vehicle.

Table 3.1 Blockchain operations

Operation	**Sender**	**Transaction**
Registration	Authentication Party	<VIN, PID> <PID, PK, Status, Misbehavior Report>
Misbehavior Report	Authentication Party	<PID, RSUID, ++Misbehavior Report>
Revocation	(automatically invoked)	<PID, Status>
Readmission	Authentication Party	<*VIN, PID> <PID, PK, Status, Misbehavior Report>

Operation	Sender	Transaction
Query_Valid_Vehicles	RSU	<select the vehicle with a PID and ‘valid’ status>

These operations are mentioned in detail with their transaction format in Table 3.1. The Registration transaction records the Vehicle Identification Number (VIN) and assigns a set of pseudo IDs (PIDs). Further, the set of public keys (PKs) generated using ECDSA are also stored on the blockchain with the PID, status of the vehicle, and misbehavior report. During registration, each vehicle is recorded with ‘valid’ as Status and is assigned zero as Misbehavior Reports.

In the Misbehavior Report transaction, for the current vehicle’s PID, the Misbehavior Report count is incremented, and the RSU that reported it is also recorded. If the misbehavior reports exceed more than three (for example), then the Revocation transaction is called automatically. This transaction updates the current vehicle’s Status as ‘invalid’. For the Readmission transaction, the vehicle’s owner goes to the nearest authority, and the authority updates the ledger to indicate the Status of the current vehicle as ‘valid’. The current vehicle is indicated by *VIN. It also updates the vehicle to have a set of new PIDs, and public/private key pairs. The Misbehavior Report for that vehicle is also updated to zero. The query, Query_Valid_Vehicles, is used by the RSUs to retrieve a list of valid PIDs of the vehicles in the blockchain. This list is checked when the vehicles request the RSUs to authenticate another vehicle’s PIDs.

As shown in Figure 3.2, the RSUs are only provided access to query the ledger and retrieve the valid PIDs and PKs of the vehicles in the network. This ensures a light-weight authentication scheme as compared to the traditional PKI approach. Further, Figure 3.2 shows the different transactions performed by the Authentication Party in the blockchain like Registration, Misbehavior Report, and Readmission.

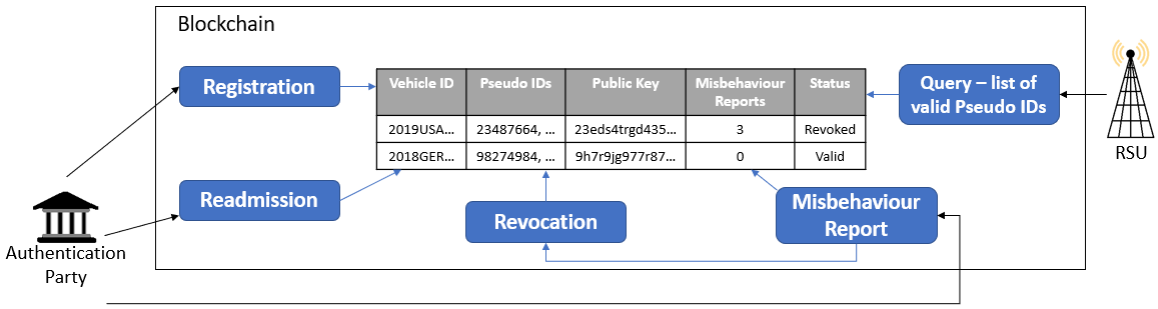

Figure 3.2 Different operations in the proposed framework

3.4 High Level Outline of Authentication

Figure 3.3 shows the flowchart used in our proposed method for authenticating the sender of a message. Each vehicle in the network uses its pseudo ID (PID) to indicate the sender of the message and attaches the public key (PK) and the digital signature of the message generated using its private key.

The receiving vehicles use the sender's PID to authenticate the messages received. Each vehicle in the network maintains a small list of valid PIDs and corresponding PKs with an expiration time for its nearby nodes. If the sender's PID is not present in this list, then the vehicle will request the nearby RSU to validate the sender's PID and PK.

The RSUs being a part of the blockchain can request a query to check if a vehicle is valid in the World State database or not. In this way, authentication of vehicles is a quick look-up for its PID in the retrieved query. Thus, we have a light-weight authentication mechanism.

Further, the digital signature of the message is verified using the public key attached along with the message. The sender of the BSM is authenticated if its PID and the digital signature of the message is validated successfully.

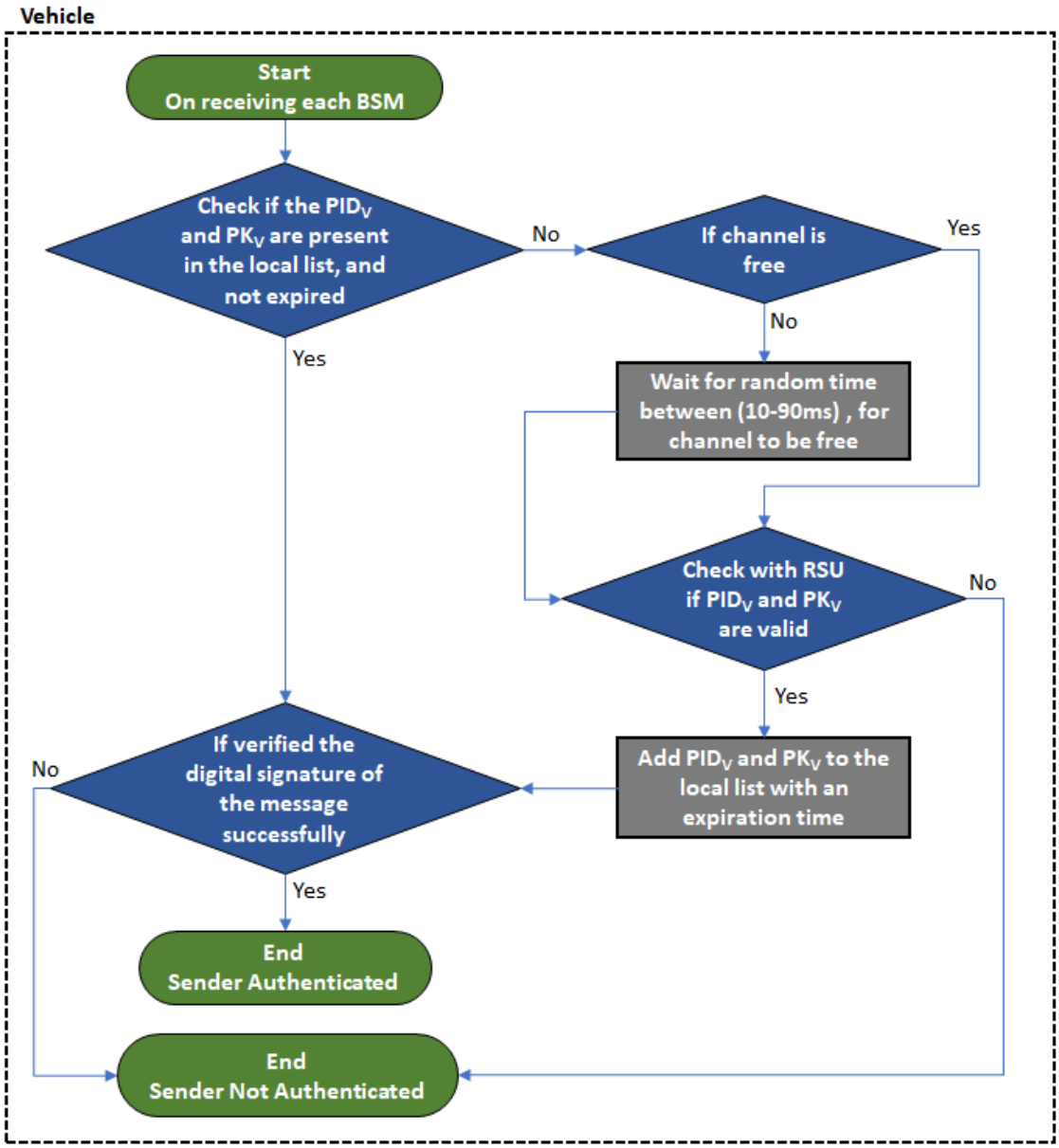

Figure 3.3 Flowchart of authentication process in vehicles

Chapter 4

4. Results and Simulation

4.1 Simulation

In our research we have used several open source software to simulate the road traffic network and to record different parameters. To simulate the network, we have used OMNET++ 5.3 [40], which is an extensible, modular, component-based C++ network simulator. We have used SUMO 0.32.0, which will provide road traffic simulation package for large road networks [41]. Further, we have used Veins 4.7.1 [42] to connect the network simulator, OMNET++, and the road traffic simulator, SUMO, to provide inter vehicle communication.

To implement the permissioned blockchain framework, we have used Hyperledger Composer [43], which is an open source framework for developing blockchain applications. Hyperledger Composer will support the Hyperledger Fabric blockchain as shown in Figure 4.1. The Model File will contain all the assets, participants and the transaction types in the network. The Script File will contain the logic for the transactions in the blockchain. The Access Control File will specify the access provided for each participant of the blockchain, and the Query file will include the queries that we need for our blockchain framework. The code for the respective files is given in APPENDIX A.

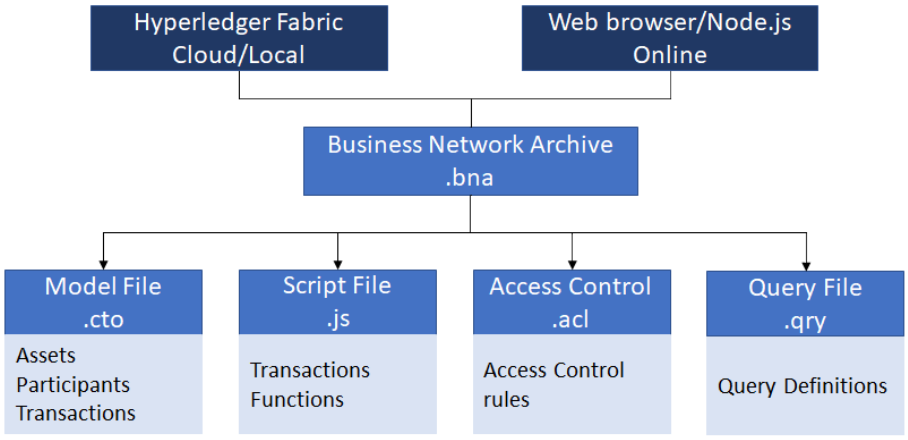

Figure 4.1 Structure of Hyperledger Composer [44]

Hyperledger Composer provides a REST API for web applications. In order to connect the OMNET++ application to the Hyperledger Composer REST API, we have used the cpprestsdk [45] external library. Moreover, we have also used Crypto++ [46] library to implement the PKI architecture with Elliptic Curve Digital Signature Algorithm (ECDSA). We have used this to compare our proposed model. We have also used the ECDSA public and private keys using Crypto++ in our proposed method.

In our proposed framework, we have two types of messages being transmitted: Basic Safety Messages (BSMs) and Wave Service Advertisements (WSAs). The parameters that we examine to compare our method with the traditional PKI architecture, are the delay due to authentication and the channel busy time. Table 4.1 shows the computer setup for our simulation.

Table 4.1 Experimentation Setup

CPU	Intel® Core™ i7-6700 CPU @ 3.40GHz
CPU-cache	8512 KB
RAM	7.7 GB

4.1.1 Simulation Setup

Table 4.2 Simulation Parameters

Simulation time	150 seconds
Frequency	5.9 GHz
No. of nodes	50
Size of ground	2500 m
Physical Layer	IEEE 802.11p
Mac Layer	IEEE 1609.4
Measured Parameters	Delay due to Authentication, Channel Busy Time, BSM Packet Size, Message Overhead

The SUMO route file included within the OMNET++ application will indicate how the simulation will behave. The route file for our simulation is included in APPENDIX B. It will indicate how frequently the nodes are being created, the path they will traverse, and the map used. The ini file in OMNET++ will indicate the simulation parameters. Table 4.2 indicates the simulation parameters that we have used. APPENDIX C shows the ini file that we have used in our simulation.

In our simulation, nodes are added every 3 seconds. Once a vehicle is generated in SUMO, a daemon process, sumo-launchd.py, will continuously listen to the requests and generate the corresponding node in OMNET++. Vehicles are generated with a maximum speed of 50 km/hr and will be generated from one point and will keep travelling through the road at maximum speed till they reach the end of the road. If the vehicle encounters traffic, it either slows down or takes an alternate route. Once the road ends, the vehicle stops transmitting messages and the finish() function is called, which is used for data collection.

4.1.2 Simulation Runs

In our simulation we have used one scenario. Figure 4.2 shows the map that is used for our simulation. We have recorded the parameters for our simulation at different times when the nodes generated were 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50.

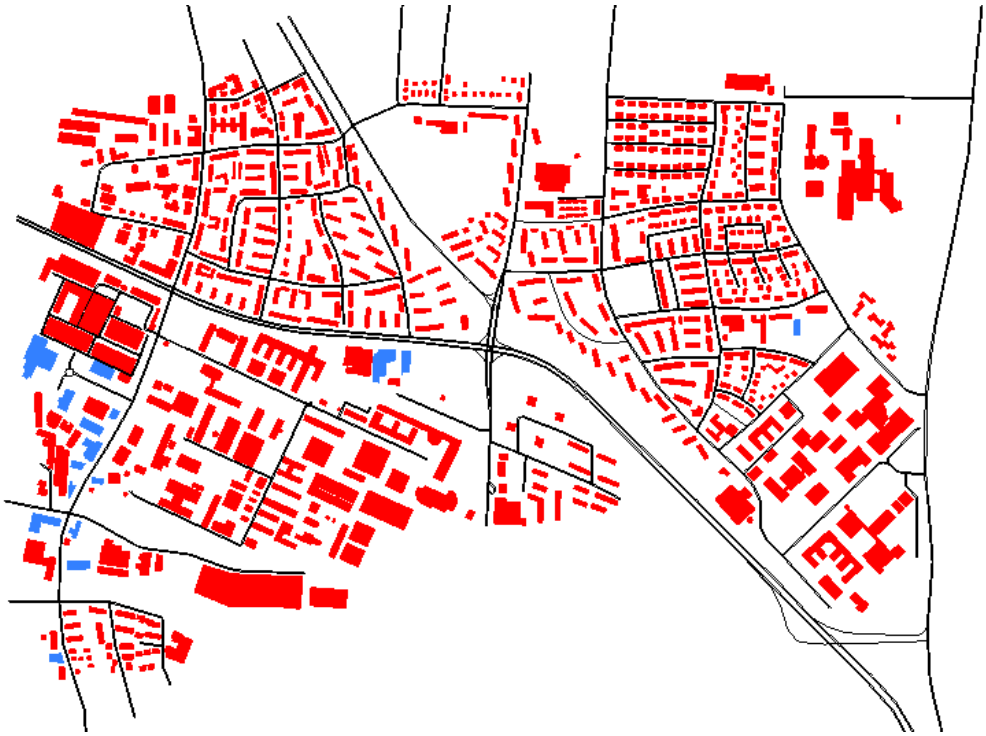

Figure 4.2 Scenario: University of Erlangen-Nuremberg, Germany

To compare our proposed method, we have simulated the same scenario with that of a traditional PKI framework in VANET as discussed in section 2.2. In our proposed method, each vehicle on receiving a BSM will check if the PID_V and the PK_V is present in the local list with valid expiration time. If not, then the RSU is contacted to verify the PID_V and PK_V if the channel is free. If the channel is busy, then the node waits for some time (we used 5 seconds) and then verifies the PID_V and PK_V with the RSU. After verifying the PID_V and PK_V, the digital signature of the message is validated using the PK_V. The RSU has access

to the blockchain to query and validate the PID and PK of vehicles. Both the approaches are examined for the delay that was caused due to the authentication framework and the channel busy time. Further, we have compared the BSM packet size in both the approaches and the additional message overhead in our approach for vehicles to request RSU services and the RSUs response back to the vehicle that is broadcasted as a WSA (WAVE Service Advertisement).

4.2 Result

Here, we analyse the results of our simulation. We have compared our proposed method to the traditional PKI framework and recorded the delay that was caused due to authentication, the channel busy time, and the BSM packet size difference and the additional message overhead in our proposed method.

4.2.1 Delay in Authentication

To study the efficiency of our proposed method, we recorded the delay in real-time for authentication. This is the time spend in seconds by each vehicle in the network receiving a BSM, to authenticate the sender of the BSM. We calculated this by using the clock() function provided by the ctime library. The summation of the duration for authenticating each BSM in the simulation is recorded when the vehicles on the road are 5, 10, up to 50 vehicles.

We calculated the average delay caused due to authentication, per BSM, by dividing the summation of the delay for authenticating all the BSMs (for simulation time – 150 seconds when the number of vehicles is 50) by the total number of BSMs transmitted in the network.

As shown in Figure 4.3, we can see that the average delay per BSM to authenticate the sender is approximately 1.9 ms using the proposed method, whereas using the PKI framework, it is approximately 3.6 ms. This is due to the additional computational time required for validating two digital signatures in the PKI approach, one for the PK_V in the certificate, and the other for the transmitted message. In our proposed method, this computation time is reduced to half by using RSU services to validate the PK_V using blockchain, and then verifying the digital signature in the message. Thus, our proposed method provides a light-weight authentication framework using blockchain.

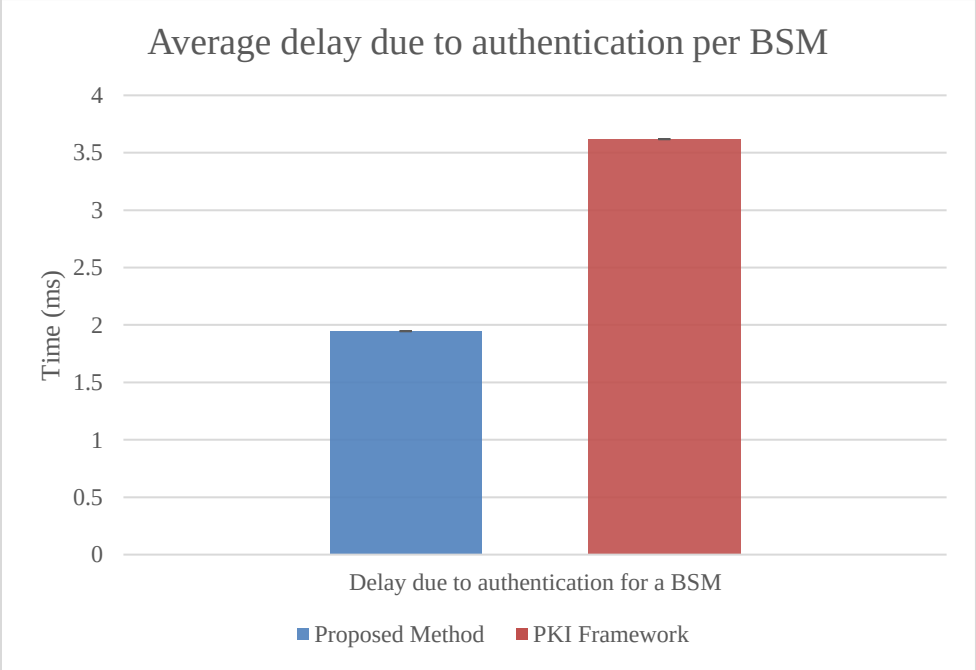

Figure 4.3 Average delay in authentication for a BSM

Further, we recorded the total delay caused due to authentication, for all BSMs, in the network as the vehicles increase in the simulation. As shown in Figure 4.4, we see that the total delay for authenticating all the BSMs in our proposed method is approximately half the delay in the traditional PKI framework. Thus, our proposed method will effectively reduce the computational time required for authentication by half of that in traditional PKI approach.

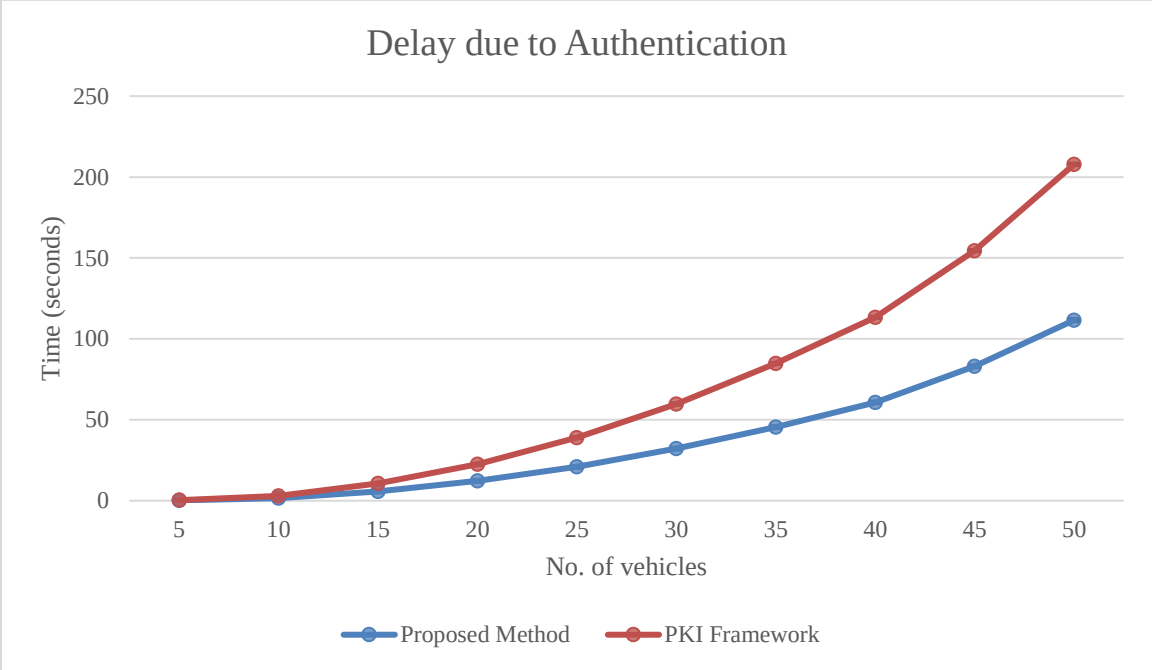

Figure 4.4 Total Delay due to authentication

4.2.2 Channel Busy Time

The channel busy time is the amount of time the MAC layer was busy due to congestion. The totalBusyTime is a scalar value recorded by Veins 4.7.1 framework and dividing it by the total simulation time will give the channel busy time in seconds.

As shown in Figure 4.5, we see that the proposed method has higher channel busy time of 0.11 seconds at simulation time 150 seconds, when compared to the PKI approach, which has 0.05 seconds. This is because of the additional communication to the RSUs, required to validate the PID and the PK of the senders. Additionally, the RSUs will broadcast a WSA when a vehicle requests to validate PID and PK. This has resulted in the increase in the channel busy time.

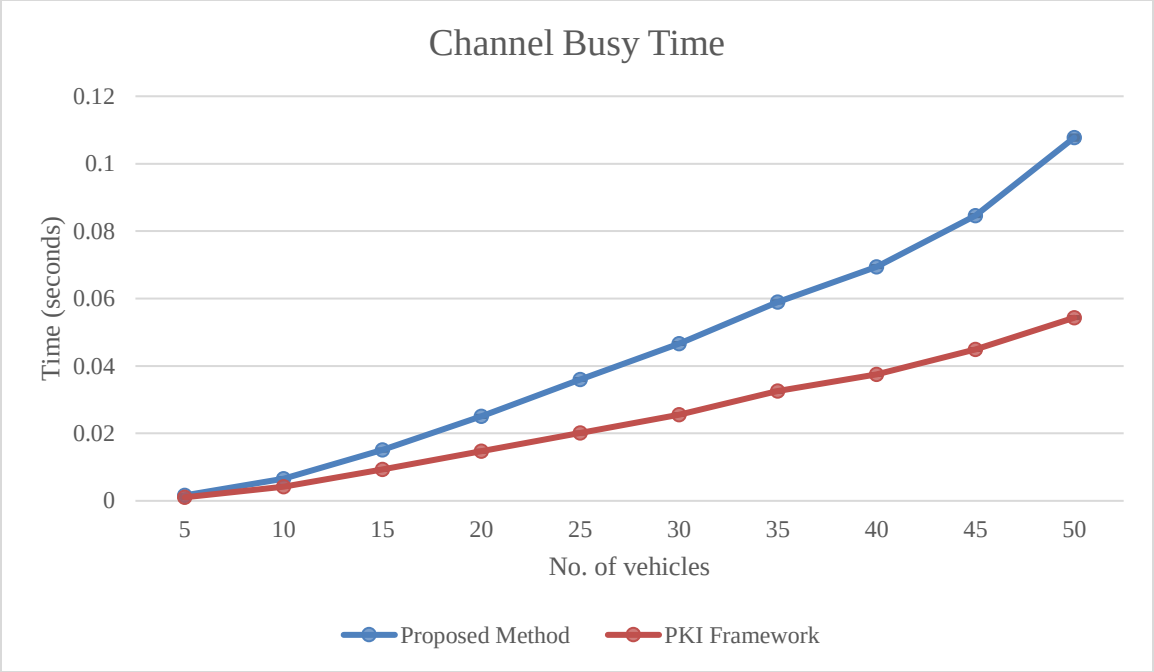

Figure 4.5 Channel busy time

4.2.3 BSM Packet Size

When comparing the BSM packet size in both the approaches, the PKI framework will have an additional certificate when compared to the size of the BSM in our proposed approach. In the simulations, the BSM packet size for the PKI approach is 235 bytes, including the digital signature of the message (64 bytes), and the certificate that contains the public key of the sending vehicle (65 bytes) and the digital signature of it (64 bytes). However, when compared to our proposed approach, the size of the BSM packet is 171 bytes. The BSM in our blockchain approach does not require the 64-byte digital signature of the public key. It contains the PID of the vehicle within the BSM and additionally the signature of the message (64 bytes) and the public key of the vehicle (65 bytes). This is shown in Figure 4.6.

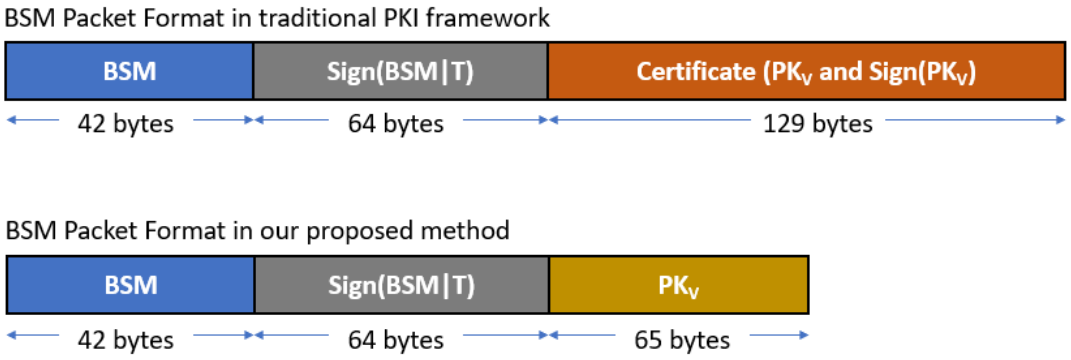

Figure 4.6 Comparison of different BSM packet size

4.2.4 Additional Messages Send

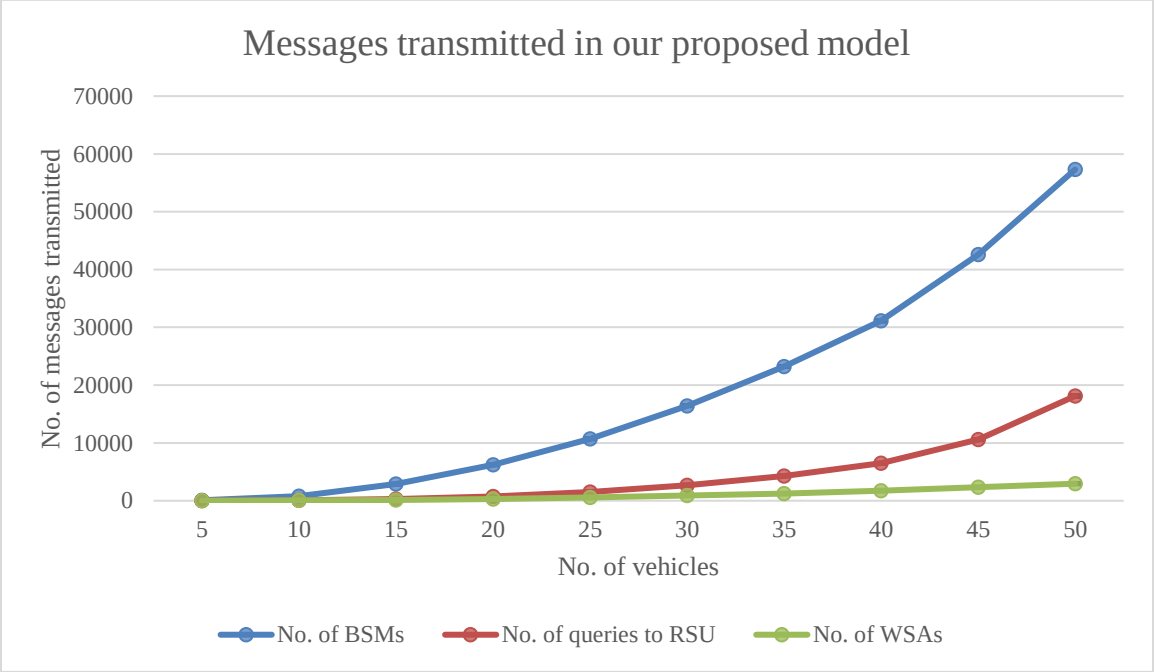

Figure 4.7 Additional messages transmitted in our proposed method

Further, we examined the additional overhead in our approach to communicate the queries to the RSU and the RSU response back to the vehicle, as a WAVE Service Advertisement (WSA), that is broadcasted to all the nearby nodes. Figure 4.7 shows the number of requests

that were send to the RSUs by the vehicles and the number of WSAs that was broadcasted to the vehicles and the number of BSMs in the network.

Chapter 5

5. Conclusion and Future Work

5.1 Conclusion

The motivation to provide a light weight authentication framework that was computationally efficient compared to the traditional PKI architecture, is realized using the blockchain framework. In our proposed framework, we validate the PID and PK of the vehicles sending messages using the RSU services. The vehicles also maintain a short list of recently validated vehicles (PID and PK) with an expiration time. After validating the PID and the PK of the sending vehicle, we then validate the digital signature of the BSM along with the timestamp that the message was send. The RSUs will have access to the blockchain and will query it to validate the PID and PK of the vehicles.

Our proposed method reduces the computational time for authentication, but this is done by sacrificing the channel busy time. Our proposed method will require additional messages to be transmitted to the RSUs and further, from the RSUs to the vehicles. Hence, this results in the additional channel busy time. However, we were able to reduce the delay due to authentication by half of that in the PKI framework. Further, using blockchains will enable a decentralized and distributed system for VANET, avoiding single point of failure.

5.2 Future Work

Channel congestion is a drawback in our proposed approach as mentioned in section 4.2.2. There are many approaches to control and improve the channel congestion. Some of these approaches can be used to check if this will reduce the channel congestion in the blockchain framework.

Further, reaching consensus is a challenging task in blockchains. There are various consensus algorithms that are quick and efficient and can be used to improve the proposed framework. Moreover, detecting a misbehaving node and reporting it to the Authentication Party requires more research. More studies need to be done to find efficient ways to detect the compromised nodes and the attackers in the network and revoke them from the network.

REFERENCES

[1] "Global status report on road safety 2018.," 2019, June 27.

[2] "First of its kind CAA study identifies Canada's worst traffic bottlenecks.," 2017, January 11.

[3] S. Yousefi, M. M. S. and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," *6th International Conference on ITS Telecommunications,* pp. 761-766, June 2006.

[4] "VANET [jpg]," 2015.

[5] R. Abassi, "VANET security and forensics: Challenges and opportunities," *Wiley Interdisciplinary Reviews: Forensic Science, 1(2), e1324,* 2019.

[6] M. N. Mejri, J. Ben-Othman and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Vehicular Communications, 1(2), 53-66,* 2014.

[7] N. Malik, P. Nanda, A. Arora, X. He and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," *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),* pp. 674-679, August 2018.

[8] X. Liu, Z. Fang and L. Shi, "Securing Vehicular Ad Hoc Networks," in *IEEE,* 2007.

[9] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, p. 217–241, August 2012.

[10] B. Cronin, "Vehicle Based Data and Availability," October 2012. [Online]. Available: https://www.its.dot.gov/itspac/october2012/PDF/data_availability.pdf. [Accessed 12 July 2019].

[11] J. Pan, J. Cui, L. Wei, Y. Xu and H. Zhong, "Secure data sharing scheme for VANETs based on edge computing," *EURASIP Journal on Wireless Communications and Networking*, December 2019.

[12] N. Bauerle, "What is the Difference Between Public and Permissioned Blockchains?," Coindesk, [Online]. Available: https://www.coindesk.com/information/what-is-the-difference-between-open-and-permissioned-blockchains. [Accessed 06 November 2019].

[13] "Introduction – Hyperledger Fabric," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/blockchain.html. [Accessed 10 August 2019].

[14] J. Lansford, J. B. Kenney and P. Ecclesine, "Coexistence of unlicensed devices with DSRC systems in the 5.9 GHz ITS band," in *2013 IEEE Vehicular Networking Conference*, Boston, MA, USA, 2013.

[15] J. Li, H. Lu and M. Guizani, "ACPN: A Novel Authentication Framework with Conditional Privacy-Preservation and Non-Repudiation for VANETs," in *IEEE Transactions on Parallel and Distributed Systems*, 2015.

[16] I. A. Sumra, H. B. Hasbullah and J. B. AbManan, "Attacks on Security Goals (Confidentiality, Integrity, Availability) in VANET: A Survey," in *Laouiti A., Qayyum A., Mohamad Saad M. (eds) Vehicular Ad-hoc Networks for Smart Cities. Advances in Intelligent Systems and Computing*, Singapore, 2015.

[17] M. Raya and J.-P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *SASN '05 Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, Alexandria, VA, USA, 2005.

[18] V. Nampally, M. R. Sharma and A. Ananthanarayanan, "A Survey on Secure Clustering Approaches for VANET," in *Scientific Figure on ResearchGate*, 2017.

[19] EdX, "Module 1: Introduction to Blockchain Components," in *LinuxFoundationX - LFS170x: Blockchain: Understanding Its Uses and Implications [Class lecture slide]*.

[20] Aziz, "Public vs Private Blockchain: What's The Difference?," masterthecrypto, [Online]. Available: https://masterthecrypto.com/public-vs-private-blockchain-whats-the-difference/. [Accessed 6 November 2019].

[21] J. Frankenfield, "Consensus Mechanism (Cryptocurrency)," Investopedia, 25 June 2019. [Online]. Available: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp. [Accessed 6 November 2019].

[22] J. Frankenfield, "Bitcoin," Investopedia, 26 October 2019. [Online]. Available: https://www.investopedia.com/terms/b/bitcoin.asp. [Accessed 6 December 2019].

[23] A. Rosic, "What is Ethereum? [The Most Updated Step-by-Step-Guide!]," Blockgeeks, 2016. [Online]. Available: https://blockgeeks.com/guides/ethereum/. [Accessed 6 December 2019].

[24] "What Is Ripple. Everything You Need To Know," Cointelegraph, [Online]. Available: https://cointelegraph.com/ripple-101/what-is-ripple. [Accessed 6 November 2019].

[25] K. Rilee, "Understanding Hyperledger Fabric — Byzantine Fault Tolerance," Medium, 14 February 2018. [Online]. Available: https://medium.com/kokster/understanding-hyperledger-fabric-byzantine-fault-tolerance-

cf106146ef43##targetText=The%20Hyperledger%20Fabric%20architecture%20lets,the%20presence%20of%20malicious%20actors.. [Accessed 6 November 2019].

[26] S. Voshmgir, "Smart Contracts," Blockchainhub Berlin, July 2019. [Online]. Available: https://blockchainhub.net/smart-contracts/. [Accessed 7 November 2019].

[27] "Proof of authority," Wikipedia, January 2018. [Online]. Available: https://en.wikipedia.org/wiki/Proof_of_authority. [Accessed 6 December 2019].

[28] B. Leiding, P. Memarmoshrefi and D. Hogrefe, "Self-managed and Blockchain-based Vehicular Ad-hoc Networks," in *UBICOMP/ISWC '16 ADJUNCT*, Heidelberg, Germany, 2016.

[29] C. Dai, X. Xiao, Y. Ding, L. Xiao, Y. Tang and S. Zhou, "Learning Based Security for VANET with Blockchain," in *2018 IEEE International Conference on Communication Systems (ICCS)*, Chengdu, China, 2018.

[30] D. J. Rankin and F. Eggimann, "The evolution of judgement bias in indirect reciprocity," *Proceedings of the Royal Society B*, 2009.

[31] N. Lasla, M. Younis, W. Znaidi and D. B. Arbia, "Efficient Distributed Admission and Revocation Using Blockchain for Cooperative ITS," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Paris, France, 2018.

[32] Z. Lu, Q. Wang, G. Qu and Z. Liu, "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 (TrustCom/BigDataSE)*, New York, USA, 2018.

[33] "Proof of Authority: consensus model with Identity at Stake.," POA Network, 11 November 2017. [Online]. Available: https://medium.com/poa-network/proof-of-authority-consensus-model-with-identity-at-stake. [Accessed 6 November 2019].

[34] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah and Z. Sun, "Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems," in *IEEE Internet of Things Journal*, 2017.

[35] T. Jiang, H. Fang and H. Wang, "Blockchain-Based Internet of Vehicles: Distributed Network Architecture and Performance Analysis," in *IEEE Internet of Things Journal*, 2019.

[36] R. Wang, J. He, C. Liu, Q. Li, W.-T. Tsai and E. Deng, "A Privacy-Aware PKI System Based on Permissioned Blockchains," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018.

[37] K. Decoster and D. Billard, "HACIT: A Privacy Preserving and Low Cost Solution for Dynamic Navigation and Forensics in VANET," in *In Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, 2018.

[38] N. Naziridis, "Comparing ECDSA vs RSA," SSL.com, 8 June 2018. [Online]. Available: https://www.ssl.com/article/comparing-ecdsa-vs-rsa/. [Accessed 9 December 2019].

[39] "What is a vehicle identification," AutoCheck, [Online]. Available: https://www.autocheck.com/vehiclehistory/vin-basics. [Accessed 10 November 2019].

[40] "OMNET++," OMNET++, 9 November 2019. [Online]. Available: https://omnetpp.org/. [Accessed 26 December 2019].

[41] "Simulation of Urban MObility," SUMO, 10 December 2019. [Online]. Available: https://sumo.dlr.de/docs/index.html. [Accessed 26 December 2019].

[42] "Vehicles in Network Simulation," Veins, 12 December 2019. [Online]. Available: https://veins.car2x.org/. [Accessed 26 December 2019].

[43] "Hyperledger Composer," Hyperledger Composer, 29 August 2019. [Online]. Available: https://hyperledger.github.io/composer/latest/. [Accessed 26 December 2019].

[44] "Welcome to Hyperledger Composer," Hyperledger Composer, 29 August 2019. [Online]. Available: https://hyperledger.github.io/composer/latest/introduction/introduction.html. [Accessed 26 December 2019].

[45] "microsoft/cpprestsdk," GitHub, 20 December 2019. [Online]. Available: https://github.com/microsoft/cpprestsdk. [Accessed 26 December 2019].

[46] "Crypto++® Library 8.2," Crypto++, 28 April 2019. [Online]. Available: https://www.cryptopp.com/. [Accessed 26 December 2019].

[47] I. Saini, S. Saad and A. Jaekel, "Identifying vulnerabilities and attacking capabilities against Pseudonym Changing Schemes in VANET," in *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, 2018.

APPENDIX A

Hyperledger Composer Files

Model File:

```
namespace org.example.basic

asset VehicleReg identified by VIN {
  o       String VIN
  o       String PID
  o       String PID2
  o       String PID3
}

asset Vehicle identified by PID {
  o       String PID
  o       String PK
  o        String status
  o       Integer misbehaviourRpt
}

participant AuthenticationParty identified by authID {
  o String authID
  o String name
}
```

```
participant RSU identified by RSUID {
  o String RSUID
  o String name
}
transaction Misbehaviour {
  --> Vehicle car
  o String RSUID
}
transaction Revocation {
  --> Vehicle car
}
transaction Readmission {
  --> VehicleReg car
  o Long pseudo1
  o       Long pseudo2
  o       Long pseudo3
}
```

Script File:

```
'use strict';
/**
```

```
 * Sample transaction processor function.
 * @param {org.vanet1.mynetwork.Misbehavior} mb The sample transaction instance.
 * @transaction
 */
async function Misbehavior(mb) {  // eslint-disable-line no-unused-vars
   mb.car.misbehaviorRpt++;
        // If the misbehaviour report is more than 3 then the vehicle is revoked.
        if(mb.car.misbehaviorRpt >= 3){
                mb.car.status = "revoked";
   }
        // Get the asset registry for the asset.
   const assetRegistry = await getAssetRegistry('org.vanet1.mynetwork.Vehicle');
   // Update the asset in the asset registry.
   await assetRegistry.update(mb.car);
}
/**
 * Sample transaction processor function.
 * @param {org.vanet1.mynetwork.Revocation} rv The sample transaction instance.
 * @transaction
 */
async function Revocation(rv) {  // eslint-disable-line no-unused-vars
```

```
    rv.car.status = "revoked";
    // Get the asset registry for the asset.
    const assetRegistry = await getAssetRegistry('org.vanet1.mynetwork.Vehicle');
    // Update the asset in the asset registry.
    await assetRegistry.update(rv.car);
}
/**
 * Sample transaction processor function.
 * @param {org.vanet1.mynetwork.Readmission} ra The sample transaction instance.
 * @transaction
 */
async function Readmission(ra) {  // eslint-disable-line no-unused-vars
    ra.car.PID = ra.pseudo1;
    ra.car.PID2 = ra.pseudo2;
    ra.car.PID3 = ra.pseudo3;
    // Get the asset registry for the asset.
    const assetRegistry = await getAssetRegistry('org.vanet1.mynetwork.VehicleReg');
    // Update the asset in the asset registry.
    await assetRegistry.update(ra.car);
}
```

Access Control File:

```
rule EverybodyCanReadEverything {
    description: "Allow all participants read access to all resources"
    participant: "org.vanet1.mynetwork.*"
    operation: READ
    resource: "org.vanet1.mynetwork.*"
    action: ALLOW
}
rule AuthPartyCanSubmitTransactions {
    description: "Allow all participants to submit transactions"
    participant: "org.vanet1.mynetwork.AuthParty"
    operation: CREATE
    resource: "org.vanet1.mynetwork.*"
    action: ALLOW
}
rule AuthPartyCanUpdateTransactions {
    description: "Allow all participants to submit transactions"
    participant: "org.vanet1.mynetwork.AuthParty"
    operation: UPDATE
    resource: "org.vanet1.mynetwork.*"
    action: ALLOW
}
```

```
rule SystemACL {
    description: "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}
rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

Query File:

```
query selectAllValidVehicles {
  description: "Select all vehilces that have valid status"
  statement:
      SELECT org.vanet1.mynetwork.Vehicle
          WHERE (PID == _$PIDParam AND status =='valid')
}
```

APPENDIX B

SUMO Configuration

SUMO Route Configuration:

```
<?xml version="1.0"?>

<routes>

<vType color="1,1,0" maxSpeed="14" minGap="2.5" length="2.5" sigma="0.5" decel="4.5" accel="2.6" id="vtype0"/>

<route id="route0" edges="-39539626 -5445204#2 -5445204#1 113939244#2 -126606716 23339459 30405358#1 85355912 85355911#0 85355911#1 30405356 5931612 30350450#0 30350450#1 30350450#2 4006702#0 4006702#1 4900043 4900041#1"/>

<flow id="flow0" number="195" period="3" begin="0" route="route0" type="vtype0"/>

</routes>
```

SUMO Configuration:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">

    <input>

        <net-file value="erlangen.net.xml"/>

        <route-files value="erlangen.rou.xml"/>
```

```
        <additional-files value="erlangen.poly.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="1000"/>
        <step-length value="0.1"/>
    </time>
    <report>
        <no-step-log value="true"/>
    </report>
    <gui_only>
        <start value="true"/>
    </gui_only>
</configuration>
```

APPENDIX C

OMNET Configuration

```
[General]

cmdenv-express-mode = true

cmdenv-autoflush = true

cmdenv-status-frequency = 1s

**.cmdenv-log-level = info

ned-path = .

image-path = ../../images

network = RSUExampleScenario

##########################################################

#          Simulation parameters                  #

##########################################################

debug-on-errors = true

print-undisposed = true

sim-time-limit = 149s

**.scalar-recording = true

**.vector-recording = false

**.debug = false

**.coreDebug = false
```

```
*.playgroundSizeX = 2500m

*.playgroundSizeY = 2500m

*.playgroundSizeZ = 50m

##############################################################

# Annotation parameters                                      #

##############################################################

*.annotations.draw = true

##############################################################

# Obstacle parameters                                        #

##############################################################

*.obstacles.debug = false

*.obstacles.obstacles = xmldoc("config.xml",
"//AnalogueModel[@type='SimpleObstacleShadowing']/obstacles")

##############################################################

#          TraCIScenarioManager parameters                   #

##############################################################

*.manager.updateInterval = 1s

*.manager.host = "localhost"

*.manager.port = 9999

*.manager.autoShutdown = true
```

```
*.manager.launchConfig = xmldoc("erlangen.launchd.xml")

##########################################################

#                  RSU SETTINGS                  #

##########################################################

*.rsu[0].mobility.x = 2000

*.rsu[0].mobility.y = 2000

*.rsu[0].mobility.z = 3

*.rsu[*].applType = "TraCIDemoRSU11p"

#*.rsu[*].applType = "MyVeinsApp"

*.rsu[*].appl.headerLength = 80 bit

*.rsu[*].appl.sendBeacons = false

*.rsu[*].appl.dataOnSch = false

*.rsu[*].appl.beaconInterval = 1s

*.rsu[*].appl.beaconUserPriority = 7

*.rsu[*].appl.dataUserPriority = 5

##########################################################

#          11p specific parameters                 #

#                NIC-Settings                      #

##########################################################

*.connectionManager.sendDirect = true

*.connectionManager.maxInterfDist = 2600m
```

```
*.connectionManager.drawMaxIntfDist = false

*.**.nic.mac1609_4.useServiceChannel = false

*.**.nic.mac1609_4.txPower = 20mW

*.**.nic.mac1609_4.bitrate = 6Mbps

*.**.nic.phy80211p.sensitivity = -89dBm

*.**.nic.phy80211p.useThermalNoise = true

*.**.nic.phy80211p.thermalNoise = -110dBm

*.**.nic.phy80211p.decider = xmldoc("config.xml")

*.**.nic.phy80211p.analogueModels = xmldoc("config.xml")

*.**.nic.phy80211p.usePropagationDelay = true

*.**.nic.phy80211p.antenna = xmldoc("antenna.xml", "/root/Antenna[@id='monopole']")

##########################################################

#                WaveAppLayer                    #

##########################################################

*.node[*].applType = "TraCIDemo11p"

*.node[*].appl.headerLength = 80 bit

*.node[*].appl.sendBeacons = false

*.node[*].appl.dataOnSch = false

*.node[*].appl.beaconInterval = 1s

##########################################################

#                Mobility                    #
```

```
##########################################################

*.node[*].veinsmobilityType.debug = true

*.node[*].veinsmobility.x = 0

*.node[*].veinsmobility.y = 0

*.node[*].veinsmobility.z = 1.895

*.node[*0].veinsmobility.accidentCount = 1

*.node[*0].veinsmobility.accidentStart = 75s

*.node[*0].veinsmobility.accidentDuration = 50s

[Config Default]

[Config WithBeaconing]

*.rsu[*].appl.sendBeacons = true

*.node[*].appl.sendBeacons = true

[Config WithChannelSwitching]

*.**.nic.mac1609_4.useServiceChannel = true

*.node[*].appl.dataOnSch = true

*.rsu[*].appl.dataOnSch = true
```

APPENDIX D

Scalar Results

Following tables show the values that are depicted as results in section 4.2, that were calculated at 95% confidence interval:

Table D. 1 Scalar results for the proposed method

No. of Vehicles	No. of BSMs transmitted	Delay (s) at the RSU for authentication (Proposed Method)	Channel Busy Time
5	80 ± 0	0.1932882 ± 0.00397563	0.001631± 0
10	810 ± 0	1.660304 ± 0.005156857	0.006588 ± 0
15	2882 ± 0	5.685357 ± 0.008248879	0.015067 ± 0
20	6206 ± 0	12.1682 ± 0.01181723	0.025109 ± 0
25	10702 ± 0	20.99469 ± 0.027719451	0.035964 ± 0
30	16425 ± 0	32.2156 ± 0.029478858	0.046644 ± 0
35	23236 ± 0	45.38536 ± 0.050937656	0.058999 ± 0
40	31110 ± 0	60.74598 ± 0.059365638	0.069437 ± 0
45	42593 ± 0	82.98144 ± 0.092097726	0.084559 ± 0
50	57310 ± 75.72	111.543 ± 0.218990367	0.107741 ± 0.000563567

Table D. 2 Scalar results for PKI framework

No. of Vehicles	No. of BSMs transmitted	Delay (s) at the RSU for authentication (PKI framework)	Channel Busy Time
5	80 ± 0	0.3259969 ± 0.004499773	0.001057 ± 0
10	810 ± 0	3.033661 ± 0.009265854	0.00422 ± 0

No. of Vehicles	No. of BSMs transmitted	Delay (s) at the RSU for authentication (PKI framework)	Channel Busy Time
15	2882 ± 0	10.58086 ± 0.009066658	0.009352 ± 0
20	6206 ± 0	22.5322 ± 0.028489313	0.014754 ± 0
25	10702 ± 0	38.93532 ± 0.055155	0.020124 ± 0
30	16425 ± 0	59.77887 ± 0.089139	0.025574 ± 0
35	23236 ± 0	84.7998 ± 0.126409699	0.032525 ± 0
40	31110 ± 0	113.353 ± 0.143286621	0.037502 ± 0
45	42593 ± 0	154.5038 ± 0.204529479	0.044884 ± 0
50	57424 ± 0	207.7973 ± 0.27740337	0.054312 ± 0

Table D. 3 Message overhead for the proposed method

No. of Vehicles	No. of BSMs transmitted	No. of Requests/Queries to RSU	No. of WSAs Broadcasted
5	80 ± 0	20 ± 0	14 ± 0
10	810 ± 0	90 ± 0	54 ± 0
15	2882 ± 0	288 ± 2.36537	134 ± 2.268432229
20	6206 ± 0	716 ± 5.297721148	291 ± 5.128110656
25	10702 ± 0	1514 ± 9.018416117	551 ± 8.038707739
30	16425 ± 0	2699 ± 11.65956171	880 ± 13.65147238
35	23236 ± 0	4296 ± 27.40604414	1248 ± 14.10319886
40	31110 ± 0	6480 ± 39.52529427	1737 ± 26.47883707
45	42593 ± 0	10566 ± 32.05964375	2316 ± 25.59294239
50	57310 ± 75.72	18121 ± 116.5713289	2949 ± 36.33726741

VITA AUCTORIS

NAME:	Sonia Alice George
PLACE OF BIRTH:	Abu Dhabi, UAE
YEAR OF BIRTH:	1994
EDUCATION:	Bachelor's in Computer Science and Engineering, Vijaya Vittala Institute of Technological (VVIT), Bengaluru, India, 2016
	Master of Science in Computer Science, University of Windsor, Windsor, ON, Canada, 2020