A University of Sussex PhD thesis

Available online via Sussex Research Online:

http://sro.sussex.ac.uk/

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Please visit Sussex Research Online for more information and further details.
Blockchain Based Secure Message Dissemination in Vehicular Networks

Ferheen Ayaz

Submitted for the degree of Doctor of Philosophy
University of Sussex
April 2022
Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Ferheen Ayaz
Abstract

Vehicular ad-hoc networks (VANETs) are one of the key elements in Intelligent Transportation System (ITS) to enable information exchange among vehicles and Roadside Units (RSUs) via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. With continuously increasing number of vehicles on road, there are numerous security and privacy challenges associated with VANETs. Communication among vehicles is needed to be secure and bandwidth efficient. Also, the messages exchanged between vehicles must be authentic so as to maintain a trusted network in a privacy-preserving manner. Furthermore, a sustainable economic model is required to incentivise honest and cooperative vehicles. Traditional security and privacy solutions in centralised networks are not applicable to VANETs due to its distributed nature, heterogeneity, high mobility and low latency requirements. Meanwhile, the new development of blockchain has been attracting significant interests due to its key features including consensus to evaluate message credibility and immutable storage in distributed ledger, which provides an alternative solution to the security and privacy challenges in VANETs.

This thesis aims to present blockchain solutions for the security and privacy of VANETs meeting the stringent requirements of low latency and bandwidth-efficient message dissemination. VANETs are simulated in OMNeT++ to validate the proposed solutions. Specifically, two novel blockchain consensus algorithms have been developed for message authentication and relay selection in presence of malicious vehicles. The first employs a voting based message validation and relay selection, which reduces the failure rate in message validation by 11% as compared to reputation based consensus. The second utilises federated learning supported by blockchain as a better privacy-preserving solution, which is 65.2% faster than the first voting based solution. Both approaches include blockchain-based incentive mechanisms and game theory analysis to observe strategic behaviour of honest and malicious vehicles. To further study the privacy aspect of vehicular networks, the integration of blockchain with physical layer security is also theoretically analysed in Vehicle-to-Everything (V2X) communications scenarios. The integration results in 8.2 Mbps increased goodput as compared to the blockchain solution alone.

In essence, our research work shows that blockchain can offer better control and security, as compared to centralised solutions, if properly adjusted according to the application and network requirements. Thus, the proposed solutions can provide guidelines for practically feasible application of blockchain in vehicular networks.
Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor Dr. Zhengguo Sheng, for actually teaching me how to carry out a research project and his painstaking attention to detail throughout my PhD. The research would not have been possible without his constructive advice and thorough guidance. I would also like to thank my second supervisor, Prof. Maziar Nekovee for his kind support and encouragement to present my research on multiple occasions.

I greatly acknowledge the appreciation I received during my industrial placement at Kinseed, which has given me motivation to excel in future. Thanks to IEEE for funding the industrial placement and providing various opportunities of networking with research community.

I cannot forget the unwavering support of my family over the past four years, right from finding a PhD position until the end of the journey. Thank you, mom and dad, for making me who I am today. Special thanks go to my husband Raheel Jamil for always being on my side during good and bad times. I am also grateful to my brother Saqib Ayaz, for not only his constant care and support but also for facilitating me multiple times during technical difficulties and simulations. Thanks to Hina, Sineen and Yousuf for always making me smile.
## Contents

List of Acronyms ................................................................................................................ x

List of Notations ................................................................................................................ xii

List of Tables ................................................................................................................... xvi

List of Figures ................................................................................................................. xvii

List of Publications .......................................................................................................... xix

Significant Participations .................................................................................................. xx

Chapter 1 - Introduction ...................................................................................................... 1
1.1 Overview ....................................................................................................................... 1
1.2 Motivation ..................................................................................................................... 2
1.3 Objectives ...................................................................................................................... 3
1.3.1 Understanding blockchain and its applications in vehicular networks ............... 3
1.3.2 Proposing blockchain solutions suitable to VANETs ........................................... 4
1.3.3 Integrating blockchain with other security and privacy solutions ....................... 4
1.4 Statement of Originality ............................................................................................. 5
1.5 Organisation of the Thesis ......................................................................................... 5

Chapter 2 – Background ..................................................................................................... 7
2.1 Vehicular Communications ....................................................................................... 7
2.1.1 System Model of VANET. .................................................................................... 9
2.1.2 Message types in VANET ................................................................................... 9
2.1.3 Security, Privacy and Trust in VANETs ............................................................. 10
2.2 Blockchain .................................................................................................................. 16
2.2.1 Introduction .......................................................................................................... 16
Chapter 4 – Blockchain-enabled Federated Learning (FL) for Message Dissemination in Vehicular Networks

4.1 Introduction

4.1.1 Federated Learning (FL)

4.1.2 FL meets Blockchain

4.1.3 FL in Vehicular Networks

4.1.4 Contributions

4.2 Blockchain-based FL for Message Dissemination

4.2.1 Blockchain-supported FL

4.2.2 PoFL based Blockchain-enabled Message Dissemination

4.3 Theoretical Analysis

4.3.1 Training Capacity of FL

4.3.2 Incentive Distribution Mechanism and its Analysis

4.3.3 Asymptotic Complexities

4.4 Simulated Performance Analysis

4.4.1 Simulation Setup

4.4.2 Loss Function of Global Model

4.4.3 Evaluation of Training Capacity of FL with and without Blockchain

4.4.4 Evaluation of Incentive Distribution Mechanism

4.4.5 Comparison of PoFL with other Solutions

4.5 Summary

Chapter 5 – Blockchain-enabled FD-NOMA based Vehicular Network with Physical Layer Security

5.1 Overview

5.1.1 Introduction
5.1.2 Motivation.............................................................................................................. 100
5.2 Analysis of Blockchain-enabled FD-NOMA based Vehicular Network............. 101
5.2.1 System Model .................................................................................................... 101
5.2.2 SINR................................................................................................................... 103
5.2.3 Secrecy Rate..................................................................................................... 106
5.2.4 Goodput............................................................................................................. 108
5.3 Simulated Performance Analysis ......................................................................... 109
5.3.1 SINR................................................................................................................... 109
5.3.2 Secrecy Rate..................................................................................................... 111
5.3.3 Percentage of Success Transmissions ............................................................... 112
5.3.4 Minimum Allowable Data rate ....................................................................... 113
5.3.5 Goodput............................................................................................................. 114
5.4 Summary ............................................................................................................. 115

Chapter 6 – Conclusions and Future Work ............................................................. 116
6.1 Contributions and Conclusions .......................................................................... 116
6.1.1 Security ........................................................................................................... 116
6.1.2 Privacy ............................................................................................................ 117
6.1.3 Trust ............................................................................................................... 117
6.2 Future Work ....................................................................................................... 118
6.2.1 Energy Aware, UAV Assisted Blockchain and AI for IoV ............................. 118
6.2.2 Quantum Resistant Cryptography ................................................................. 119
6.2.3 Network Layer Aspects of Blockchain ......................................................... 119
6.2.4 Alternatives to Linear Distributed Ledger ................................................. 120

Bibliography ............................................................................................................. I

Appendix A ............................................................................................................... XX
List of Acronyms

5G  Fifth-generation
A2A  Air-to-Air
A2G  Air-to-Ground
AI  Artificial Intelligence
AV  Autonomous Vehicle
AWGN  Additive White Gaussian Noise
C-V2X  Cellular Vehicle-to-Everything
CA  Central Authority
CDF  Cumulative Distribution Function
DAG  Direct Acyclic Graph
DCF  Distributed Coordination Function
DIFS  DCF Interframe Space
DR  Data rate
DSRC  Dedicated Short Range Communications
FD-NOMA  Full Duplex Non-Orthogonal Multiple Access
FL  Federated Learning
FV  Failure in Validation
GPS  Global Positioning System
IoT  Internet-of-Things
IoV  Internet-of-Vehicles
ITS  Intelligent Transportation System
LB  Lower Bound
MAC  Medium Access Control
MANET  Mobile Ad-hoc Network
MSE  Mean Squared Error
NOMA  Non-Orthogonal Multiple Access
OBU  On-Board Unit
P2P  Peer-to-peer
PBFT  Practical Byzantine Fault Tolerant Algorithm
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PLS</td>
<td>Physical Layer Security</td>
</tr>
<tr>
<td>PoET</td>
<td>Proof-of-Elapsed-Time</td>
</tr>
<tr>
<td>PoFL</td>
<td>Proof-of-Federated-Learning</td>
</tr>
<tr>
<td>PoQF</td>
<td>Proof-of-Quality-Factor</td>
</tr>
<tr>
<td>PoS</td>
<td>Proof-of-Stake</td>
</tr>
<tr>
<td>PoW</td>
<td>Proof-of-Work</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RSU</td>
<td>Road Side Unit</td>
</tr>
<tr>
<td>SIC</td>
<td>Successive Interference Cancellation</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Interframe Space</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference and Noise Ratio</td>
</tr>
<tr>
<td>SUMO</td>
<td>Simulation of Urban Mobility</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>UB</td>
<td>Upper Bound</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle-to-Infrastructure</td>
</tr>
<tr>
<td>V2N</td>
<td>Vehicle-to-Network</td>
</tr>
<tr>
<td>V2P</td>
<td>Vehicle-to-Pedestrian</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-Vehicle</td>
</tr>
<tr>
<td>V2X</td>
<td>Vehicle-to-Everything</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad-hoc Network</td>
</tr>
<tr>
<td>VeINS</td>
<td>Vehicles In Network Simulation</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wireless Access in Vehicular Environment</td>
</tr>
</tbody>
</table>
List of Notations

Distances and speed

\[ d_{i,j} \] Distance between node \( i \) and node \( j \)
\[ \text{dir}_{i,s} \] Direction of node \( i \) with respect to sender \( s \)
\[ d_{\text{neigh}}^{\min} \] Minimum distance between neighbour nodes
\[ d_{\text{hop}}^{\min} \] Minimum hop distance
\[ DF_i \] Distance Factor of node \( i \)
\[ R \] Transmission range
\[ \lambda_V \] Node/vehicle density (nodes/m\(^2\))
\[ v_i \] Speed of node \( i \)
\[ \sigma_i^2 \] Variance of \( v_i \)

Numbers and means

\[ n_{\text{hop}} \] Hop number
\[ n_{\text{th}} \] Threshold number of votes
\[ n_{\text{tr}} \] Number of simultaneous transmissions
\[ n_{\text{if}} \] Number of interference nodes
\[ n_{\text{neigh}} \] Number of neighbour nodes
\[ n_{\text{mn}} \] Number of mining nodes
\[ n_{m} \] Number of malicious nodes
\[ n_{h} \] Number of honest nodes
\[ n_{cp} \] Number of colluding players
\[ n_{e} \] Number of eavesdroppers
\[ n_{s} \] Number of senders
\[ n_{r} \] Number of receivers
\[ n_{\text{FL}} \] Number of nodes participating in FL
\[ n_B \] Number of nodes uploading local models via FL blockchain
\[ n_{WB} \] Number of nodes uploading local models without FL blockchain
\[ n_{\text{V}} \] Number of nodes (vehicles) with RSU in transmission range
\( n_v' \) Number of moving nodes reaching RSU
\( n_{RLY} \) Number of RLYs
\( n_A \) Number of acknowledgement messages
\( n_Z \) Number of possible data sizes
\( \mu_m \) Mean number of malicious nodes
\( \mu_h \) Mean number of honest nodes
\( \mu_d \) Mean distance between nodes and RSU
\( \mu_v \) Mean speed of nodes (vehicles)

### Probabilities
\( p_m \) Probability of malicious nodes
\( p_{cp} \) Probability of collusion
\( p_t \) Average transmission probability
\( p_{suc} \) Probability of success transmission
\( p_{col} \) Probability of collided transmission
\( p_{idle} \) Probability of encountering idle slot
\( p_z \) Probability of using data size \( s_z \)
\( p_{out} \) Outage probability

### Time
\( \tau_i \) Validation time of node \( i \)
\( a_{\tau_i}, b_{\tau_i} \) Upper, lower limit of \( \tau_i \)
\( t_{icd} \) Time at which incident message is received
\( T_{delay} \) Time delay to finalize consensus
\( T_{slot} \) Time slot in MAC layer
\( T_{suc} \) Time for success transmission
\( T_{col} \) Time for collided transmission
\( T_{DIFS}, T_{SIFS} \) Time intervals for DIFS, SIFS
\( T_{CTS}, T_{ACK}, T_{RTS} \) Time intervals for DCF related operations
\( T_{avg} \) Average length of a time slot in DCF
\( T_{MB} \) Time to transmit a microblock
\( T_{\text{eyp}} \) : Time to encrypt a block
\( TS \) : Time slot to upload local models

**Throughput**

\( \lambda_B \) : Blockchain throughput (block/s)
\( \lambda_{KB} \) : Keyblock throughput (keyblock/s)
\( \lambda_{MB} \) : Microblock throughput (microblock/s)
\( DR \) : Data rate (bit/s)

**Power and channel-related parameters**

\( P_{\text{noise}} \) : Noise Power
\( P_i \) : Power of signal transmitted by node \( i \)
\( QF_i \) : Quality Factor of node \( i \)
\( \text{SINR}_{i,j} \) : SINR between node \( i \) and node \( j \)
\( Q(\text{SINR}_i) \) : Quality of node \( i \)’s SINR
\( g_{i,j} \) : Channel gain between node \( i \) and node \( j \)
\( h_{i,j} \) : Channel coefficient between node \( i \) and node \( j \)
\( C_{i,j} \) : Secrecy rate between node \( i \) and node \( j \)
\( \alpha \) : Path loss exponent
\( \eta \) : Coefficient of self-interference
\( \beta_1 \) : Threshold of \( \text{SINR}_{i,j} \)
\( \beta_2 \) : Threshold of \( C_{i,j} \)

**Incentive Distribution**

\( CC \) : Call Compensation
\( CC_{mn} \) : Call Compensation for mining nodes
\( CC_r \) : Call Compensation for RLYs
\( \omega_1 \) : Weight to distribute \( CC \) among mining nodes
\( \omega_2 \) : Weight to distribute \( CC \) among mining RLYs
\( U_i \) : Utility of node \( i \)
\( s_i \) : Data size of node \( i \)
\(Cost(s_i)\) Cost of training \(s_i\)

\(I\) Incentive in FL

\(TC\) Transaction Charge

\(\rho_i\) Cost coefficient of node \(i\)

\(Rep_i\) Reputation of node \(i\)

\(Rep_T\) Reputation Threshold

\(Rep_{Rew}\) Reputation Reward

**Federated Learning**

\(LF(w^k_x)\) Loss function of model \(x\) (local or global) at \(k^{th}\) iteration

\(w^k_i\) Weights of local model by node \(i\) at \(k^{th}\) iteration

\(w^k_G\) Weights of global model at \(k^{th}\) iteration

**Miscellaneous**

\(B_i\) Behaviour of node \(i\)

\(\kappa\) Consensus parameter

\(L\) Length of a packet

\(W\) Window size

\(FV\) Failure in Validation

\(ORG\) Originator

\(RLY\) Relay node
List of Tables

Table 2.1: Comparison of consensus algorithms ............................................................. 20
Table 2.2: Advantages and challenges of blockchain applications ................................. 25
Table 2.3: VANET issues and opportunities using blockchain ................................. 29
Table 3.1: Consensus Algorithms used in VANETs ......................................................... 31
Table 3.2: Payoff Matrix $U_i, U_y$, where $U_i=$ Utility of mining node $i$ with the highest $QF_i$ and $U_y=$ Utility of any other mining node at $n_{hop} = 1$ ................................................................. 50
Table 3.3: Comparison of asymptotic complexities ......................................................... 56
Table 3.4: Simulation parameters for PoQF message dissemination ............................... 58
Table 3.5: Comparison of PoQF with other consensus algorithms for blockchain-enabled VANETs ........................................................................................................................... 68
Table 4.1: FL issues and blockchain solutions ................................................................ 72
Table 4.2: Multi-hop relay selection limitations and solutions offered by blockchain-based FL .................................................................................................................................. 72
Table 4.3: FL components and corresponding blockchain support .................................. 77
Table 4.4: Parameter-sharing required from neighbour nodes ....................................... 83
Table 4.5: Reward gained and payment made by players ................................................ 86
Table 4.6: Comparison of asymptotic complexities ......................................................... 88
Table 4.7: Simulation parameters for blockchain-based FL and message dissemination 89
Table 4.8: Loss (MSE) of global model after 100 iterations ........................................... 91
Table 4.9: $\lambda_{MB}$ and $\mu_d$ with respect to $\lambda_V$ ...................................................... 94
Table 5.1: Parameters used in simulation of FD-NOMA based vehicular network ...... 109
Table A.1: Simulation tools used for each of the proposed solution ........................... XX
Table A.2: Simulation Parameters ................................................................................. XX
List of Figures

Figure 1.1: A blockchain-based message dissemination solution........................................ 4
Figure 2.1: IoV in C-V2X communications.......................................................................... 7
Figure 2.2: System model of a VANET.............................................................................. 8
Figure 2.3: Relationship between security, privacy, and trust in VANETs......................... 14
Figure 2.4: The blockchain structure. ............................................................................. 16
Figure 2.5: The stages of PBFT algorithm.......................................................................... 20
Figure 2.6: Variations in block contents. .......................................................................... 21
Figure 2.7: A hierarchical blockchain structure................................................................. 23
Figure 2.8: Fork occurrence in conventional blockchain.................................................... 23
Figure 2.9: Parallel addition of microblocks to resolve forks in blockchain in vehicular networks ................................................................................................................................................................................. 24
Figure 3.1: Voting based message validation and dissemination.......................................... 31
Figure 3.2: The proposed PoQF consensus......................................................................... 34
Figure 3.3: Flowchart of actions by mining node at $n_{hop} = 1$. ..................................... 35
Figure 3.4: Distribution of Call Compensation................................................................... 42
Figure 3.5: Resolving forks in the proposed PoQF consensus. ........................................... 48
Figure 3.6: $FV$ with respect to $n_{mn}$. ........................................................................... 59
Figure 3.7: Setting up $CC_r$, $TC$ and $n_{max_{hop}}$ for collusion resistant incentive distribution................................. 60
Figure 3.8: Setting up $\omega_1$ and $\omega_2$ for collusion resistant incentive distribution...... 61
Figure 3.9: $T_{delay}$ with respect to $n_{mn}$. ..................................................................... 63
Figure 3.10: $\lambda_{KB}$ with respect to $p_m$. ...................................................................... 64
Figure 3.11: Average success rate with respect to speed................................................... 64
Figure 3.12: Values of $a_{t_l}$ and $b_{t_l}$. .......................................................................... 65
Figure 3.13: Comparison of PoQF with PoS and PoET at $n_{th} = \mu_m + 1$. ..................... 66
Figure 4.1: The FL process. ............................................................................................ 71
Figure 4.2: Modes of FL. ............................................................................................... 74
Figure 4.3: Modes of FL in VANET.................................................................................. 74
Figure 4.4: Blockchain-based FL and Message Dissemination.......................................... 75
Figure 4.5: The proposed stages of blockchain-supported FL........................................... 79
Figure 4.6: Flowchart of actions by node $i$ according to PoFL based blockchain-enabled message dissemination. .......................................................................................................................... 82

Figure 4.7: The flow of steps in blockchain-supported FL and PoFL based message dissemination. .................................................................................................................................................. 83

Figure 4.8: Loss (MSE) of global model with 50% adversary. ................................................................. 90

Figure 4.9: Loss (MSE) of global model after 100 iterations with malicious nodes. ............ 92

Figure 4.10: Loss (MSE) of global model after 100 iterations with selfish nodes. ............ 93

Figure 4.11: Number of nodes uploading local model in $T$. ........................................ 94

Figure 4.12: Utility of players in Stackelberg Game with equilibrium points (*). ........... 95

Figure 4.13: Message delivery ratio with 300 nodes. ............................................................... 96

Figure 4.14: Average time delay per hop. ..................................................................................... 97

Figure 5.1: The system model. ......................................................................................................... 101

Figure 5.2: $SINR_{t_j}$ and $Pr(SINR_{t_j} \leq \beta_1)$ with respect to $d_{t_j}$. ........................................ 110

Figure 5.3: $Pr(C_{t,j} \geq \beta_2)$ with respect to $d_{t_j}$. ................................................................. 111

Figure 5.4: Success transmissions in presence of jammer and eavesdroppers, $n_{itf}^j = 2, n_{itf}^k = 1$. .................................................................................................................. 112

Figure 5.5: Minimum allowable $DR$. ................................................................................... 113

Figure 5.6: Goodput in presence of jammers and eavesdroppers. ............................................. 114

Figure 6.1: Integrated solution of connected UAVs and IoV. ...................................................... 118

Figure A.1: Simulation map of University of Sussex campus. .................................................... XXI

Figure A.2: Screenshot of Veins running in OMNeT++. ............................................................ XXI
List of Publications


Significant Participations

Chapter 1 - Introduction

1.1 Overview

The extensive use of digital technologies and multiple forms of interactions, (e.g., human-to-human, human-to-device, device-to-device) have led to the development of Internet-of-Things (IoT) into a global network connecting diverse devices, for example, smartphones, TV and vehicles. Internet-of-Vehicles (IoV) is one particular vertical domain of IoT incorporating communications between vehicles and other objects, using various communication models and approaches, for example, vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X), etc. IoV offers multiple services including resource-sharing, task-offloading and infotainment, etc. However, it can be considered as an advanced form of Vehicular-Ad-hoc-Network (VANET), and its primary objective is real-time sharing of information for a safe journey and smooth traffic flow [1].

Disruptions in traffic flow and congestions have been very common with increasing vehicle density on road, leading to a high risk of accidents and a serious threat to road safety [2]. Safe driving conditions can be achieved by VANET in which a vehicle can initiate reliable and efficient message dissemination in emergency situations, such as a road accident [3].

The oncoming vehicles may re-route themselves to avoid traffic congestion by receiving an alert prior to entering into an affected area. Thus, timely exchange of messages about an incident or traffic jam can lead to a better journey for the oncoming vehicles.

Nevertheless, due to massive growth in the number of connected vehicles, VANET comes across several challenges which are yet to be addressed for its secure implementation. A VANET encompasses large number of untrusted nodes, including vehicles and Road Side Unit (RSU), some of which may be malicious and tend to disrupt message exchange during emergency. Privacy is also an essential requirement in communications among untrusted nodes. Furthermore, it is not feasible for a Central Authority (CA) to be in-charge of the entire network, as it may become a single point of failure affecting the whole network. Therefore, decentralisation is recommended for VANETs [4]. Considering decentralisation, blockchain is seen as a promising solution to some of the challenges in VANETs. Blockchain is originally a distributed ledger to record history of financial
transactions in the form of immutable and cryptographic blocks [5]. However, its robust architecture and application of its framework outside finance can potentially support VANETs. This thesis explores blockchain as a solution to distributed message dissemination in VANETs and utilises its features for resolving security, privacy and trust issues.

Despite the complementing features of blockchain, such as decentralisation, immutability and privacy support in an infrastructure-less environment, the integration of conventional blockchain in a dynamic and heterogenous vehicular network is yet another challenge. It is mainly due to latency constraints in emergency message dissemination and high speed of mobile nodes. This thesis investigates blockchain-based solutions particularly suited to a VANET environment.

1.2 Motivation

There are two main aims behind this research. First is to resolve security, privacy and trust related issues of VANETs through blockchain. Second is to address challenges of existing blockchain approaches and propose practically feasible solutions specifically for VANETs. The motivations behind utilising blockchain and proposing specific solution for VANETs are described below.

One of the security challenges of VANETs is credible message dissemination, i.e., to verify that a message is true before it is forwarded. Blockchain can be utilised to verify messages as it is conventionally used to record history of verified transactions only. Nodes undergo a mutual agreement called consensus to verify a transaction. Similarly, consensus can be utilised for message validation. Also, decentralised relay selection for efficient multi-hop communications in VANETs is another challenge. Nodes undergoing a blockchain consensus can mutually select an appropriate relay among themselves for timely dissemination of message in a distributed and trusted manner. Furthermore, blockchain-based transactions can support secure and immutable distribution of incentives among nodes. Since blockchain stores information in the form of encrypted blocks, it can also ensure privacy preservation in VANETs.
If blockchain is employed for emergency message dissemination in VANETs, it must support high throughput and low latency. Famous blockchain, such as bitcoin, uses Proof-of-Work (PoW) consensus which has a latency of around ten minutes [6]. Due to short-lived connectivity among high-speed nodes, PoW is not appropriate for VANETs. In addition, the high computation complexity and power requirements of PoW are also not suitable for On-Board Unit (OBU) on vehicles. Alternatives to PoW are proposed for achieving better throughput. For example, Proof-of-Stake (PoS) is commonly used in vehicular communications, where stakes are considered as reputation or trust ratings of nodes [7]. However, PoS is considered biased towards nodes with high stakes. The consensus algorithms with their strength, weaknesses and their suitability in VANETs are further discussed in Chapter 2. Due to limitations of existing consensus algorithms, this thesis proposes novel consensus algorithms specifically designed for VANET applications.

1.3 Objectives

This thesis mainly answers two research questions:

1) How to design an efficient message dissemination solution for vehicular networks using blockchain?
2) How blockchain can ensure the security, privacy and trust in vehicular networks?

The detailed objectives of the thesis are as follows

1.3.1 Understanding blockchain and its applications in vehicular networks

The first objective of the thesis is to gain a better understanding of blockchain, its structure, variations, consensus algorithms, features and applications in wireless networks. Next is to identify security, privacy and trust requirements of vehicular communications and analyse advantages and limitations of existing solutions, particularly focusing on why and how blockchain can be a better solution. Later, it is aimed to thoroughly analyse blockchain’s performance and suitability in vehicular networks. Therefore, the performance of blockchain-enabled vehicular networks is investigated under different conditions, i.e., varying percentage of malicious nodes, various speeds and traffic densities of nodes on road and different strategic behaviour.
1.3.2 Proposing blockchain solutions suitable to VANETs

One of the main objectives is to investigate the limitations of existing blockchain consensus algorithms and propose new solutions which are particularly appropriate to message dissemination in VANETs. The proposed consensus algorithms aim to validate a message, i.e., evaluate its credibility and select relay node in a distributed manner. The latency in message dissemination and blockchain throughput are considered as key parameters to analyse performance of the proposed consensus algorithms. Complete blockchain-based message dissemination solutions are proposed, as shown in Figure 1.1, which can validate a message initiated in case of incident or emergency and select the most appropriate relay node in a decentralised fashion. The solutions also include a sustainable economic model, in which the originator of incident pays virtual credits or cryptocurrency to the affected nodes on road, as a compensation of causing an incident.

![Figure 1.1: A blockchain-based message dissemination solution.](image)

1.3.3 Integrating blockchain with other security and privacy solutions

Another objective is to investigate how blockchain can support other security and privacy approaches of vehicular networks, e.g., Federated Learning (FL) and Physical Layer Security (PLS). Integrated solutions are proposed aiming for higher security and privacy.
Detailed theoretical and simulated analysis are conducted for performance evaluation of integrated approaches and practical feasibility of their co-existence.

1.4 Statement of Originality

The original contribution of the thesis includes substantial blockchain solutions and consensus algorithms specifically for message dissemination in VANETs. The following chapters of the thesis are believed to be containing significant and original contributions

- The first contribution of the thesis is the proposed message validation by voting and competitive relay selection on the basis of its distance from sender and other channel quality parameters. Chapter 3 theoretically analyses the performance bounds of failure and latency in message validation. An incentive mechanism to promote honest voting and relay selection is also proposed and analysed using game theory.

- As an alternative to voting, Chapter 4 proposes a faster relay selection method by using blockchain-enabled FL. An incentive mechanism is presented to motivate nodes for a cooperative FL process and is analysed using Stackelberg game model. FL also enhances privacy of the proposed solution.

- To further study privacy in blockchain-enabled vehicular networks, the last contribution of this thesis is to analyse possible integration of PLS with blockchain. Chapter 5 theoretically analyses blockchain implementation on application layer with respect to physical layer characteristics, i.e., Signal to Interference Noise Ratio (SINR) and secrecy rate.

1.5 Organisation of the Thesis

The remainder of the thesis is outlined as follows. Chapter 2 discusses the technical background of vehicular communications, security, privacy and trust in VANETs. It also provides detailed introduction to blockchain and reviews blockchain-based solutions. Important and original contributions are presented in Chapter 3, Chapter 4 and Chapter 5, including a voting based blockchain solution, a blockchain-enabled FL approach and security analysis of blockchain integrated with PLS against jamming and eavesdropping
attack, respectively. A conclusive summary of contributions and possible future research directions from this thesis are highlighted in Chapter 6. A description of tools used to obtain simulated results is present in Appendix A.
Chapter 2 – Background

This chapter consists of two main sections. First is vehicular communications, which briefly introduces different interfaces through which vehicles can communicate, describes VANET system model and discusses its security, privacy and trust requirements. Second is blockchain, which introduces blockchain structure, its features, consensus and current variations. The applications of blockchain to meet security, privacy and trust requirements are also discussed.

2.1 Vehicular Communications

Broadly, vehicular communications in IoV aims to attain full connectivity of a vehicle with not only other vehicles on road but also with every other object including passengers’ smart phones, external management platform, navigation systems and other road users for both infotainment and entertainment purposes. An IoV network is composed of an environment consisting of humans, vehicles and things exploiting vehicle-to-everything (V2X) communications. The evolution of fifth-generation (5G) has led to the introduction of Cellular V2X (C-V2X) communications, which exploits Long Term Evolution direct device-to-device communications and allows vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicle-to-pedestrian (V2P) communications without network coverage. When in coverage, the communications can be augmented by Vehicle-to-Network’s (V2N’s) extended range and assistance, as shown in Figure 2.1 [8].
The major aim of Intelligent Transportation System (ITS) is to improve road safety, driving conditions and traffic flow through vehicular communications. It relies primarily on V2V and V2I communications for sharing critical messages related to road and driving conditions [9]. A VANET consists of only two types of communications, V2V and V2I [10]. VANET predominately exploits Dedicated Short Range Communications (DSRC) for V2V, V2I and hybrid communications in a single hop or multi-hop fashion [11]. DSRC is standardised as IEEE 802.11p and IEEE 1069.x, combinedly called as Wireless Access in Vehicular Environment (WAVE) [12]. A VANET is a special class of Mobile Ad-hoc Networks (MANETs) with pre-defined routes (roads). As shown in Figure 2.2, vehicles exchange messages with nearby vehicles in a particular transmission range via V2V communications and communicate with RSU via V2I interface. Vehicles and RSUs are termed as ‘nodes’ in a VANET.

![Figure 2.2: System model of a VANET.](image)

Road safety is ensured in VANETs by exchanging appropriate messages among vehicles. Any alteration to these messages may lead to network failure and posing a serious threat to safety. Regardless of the technology, securing information is of utmost importance in V2V and V2I communications. Three main concerns in the design of secure VANETs are: security, privacy and trust. Security is defined as the approach to make the system free from danger or threat and the measures taken to be protected [13]. For example, security
in VANET can refer to a solution to prevent an attack by malicious nodes. Privacy deals with rights to access or hide identities, messages and protection of personal information such as position or heading direction of a node. Trust deals with how nodes perceive each other (such as honest or adversarial) and received messages (such as true or false) [10]. There are two types of adversarial nodes in VANETs. One is malicious node, which tends to damage the network, for example, sharing false messages or attempting to validate false messages. The other is selfish node, which tends to save its energy by not cooperating in forwarding messages in multi-hop communications [14].

2.1.1 System Model of VANET

As shown in Figure 2.2, a VANET consists of the following major components:

- **Vehicles**: The vehicles are mobile nodes containing OBUs equipped with sensors, such as Global Positioning System (GPS) and camera, processors, storage, network devices and transceivers. The vehicles can communicate with its neighbour nodes via DSRC in a particular transmission range.

- **Road Side Unit (RSU)**: RSUs are immobile nodes on the road edges dedicated to communicate with vehicles via DSRC as well as with other RSUs and CA via infrastructural network. RSUs can facilitate in emergency message dissemination and provide internet connection to vehicles.

- **Central Authority (CA)**: A CA performs administrative roles in VANET including node registration, maintaining trust, managing incentive distribution schemes and assigning or revoking cryptographic keys and certificates. CA usually communicates with RSU only via wired connection.

2.1.2 Message types in VANET

The two major types of messages broadcasted in VANET are:

- **Beacon messages**: They are single hop periodic messages regularly initiated by mobile nodes to share their status on road with their neighbours. The information exchanged in beacon messages is usually position, speed, acceleration and vehicle
type. The interval between two beacon messages generation by a same node is usually 100 ms to 1000 ms [15].

- **Emergency messages**: These messages are event-triggered and can be initiated by any node in case of unusual or emergency situations such as an incident or traffic jam. An emergency message is usually generated by vehicle if it detects a situation via camera or any other sensor equipped in OBU [16]. Because of the strict time constraint in case of emergency, an efficient routing protocol is required to transmit a message securely via multi-hop V2V or V2I communications. The issue of high dynamicity of nodes and scalability is also considered while designing routing protocol of messages. The allowable latency in communicating emergency messages depends on the type of emergency and is upper bounded by 1000 ms [3].

2.1.3 Security, Privacy and Trust in VANETs

Figure 2.3 illustrates the relationship among security, privacy and trust requirements and solutions in VANETs. This subsection discusses key attributes of each domain and reviews some solutions existing in literature.

A. **Security**

Security refers to the condition that VANET is protected from threats and attacks by adversary. The key attributes of security in VANETs include

- **Credibility**: It is essential for nodes in VANET to assess credibility of a message before dissemination. A message initiated by a node may not always be true. This is because a malicious node can affect the credibility of a message in the following ways [17]
  - generate a false message
  - validate a false message
  - deny a true message
  - collude with other nodes to perform any of the above actions

The credibility of a generated message is assessed by three approaches. The first is pre-event, where each node has its trust rating based on which its originated message is considered true or false. If its trust rating falls below a certain threshold,
the message generated by the node is ignored. This approach can be centralised where a CA stores trust ratings [18] or cluster-based where a cluster head manages trust rating [19]. The second is post-event, in which a message is validated on the basis of a threshold number of endorsements or votes from the neighbouring nodes [20]. The third is hybrid, where only trusted nodes take part in evaluation of message credibility [21]. The metrics to evaluate a message validation or credibility approach in vehicular communications include the speed of reaching a decision, communication overhead and tolerance rate of malicious nodes.

- **Availability**: It is the assurance that a network remains operational in presence of malicious nodes. In case of emergency on road, a VANET must be able to independently disseminate the message up to a required number of hops, desired area or time limit, so that the information is available to all relevant nodes. Therefore, a swift and reliable message dissemination solution is required in VANETs. The receiving nodes must be able to select the most appropriate relay node among themselves, which can effectively forward a message to a large number of other nodes [22]. Relay selection in vehicular communications plays a crucial role in message broadcasting. An inappropriate relay may cause unacceptable latency or sometimes failure in delivering a message to a desired number of vehicles or area. A central node such as RSU can be used for message dissemination. However, it requires a large infrastructure investment [23] and its volume and placement positions are crucial for system performance [24]. Moreover, RSU assisted routing protocol may result in higher latency because of an increased route discovery time [25]. Relay node selection depends on a number of factors, such as, distance between sender and receiver, and some physical and Medium Access Control (MAC) layer properties including channel quality, collision probability and SINR. Prior to relay node selection, probabilistic estimations predict the optimal node with the most suitable combination of its distance and link stability at the time of transmission [26] - [27]. However, probabilistic predictions rely on certain approximations, for example, number of nodes within a transmission range, which may not be highly accurate when varying traffic and speed limits are imposed in a vehicular network. A cross-layer approach integrating both MAC and physical
layer for optimum bandwidth utilisation is presented in [28]. The weighted combination of distance and channel quality are used to determine link stability for relay node selection in [29]. To make relay selection more adaptive to network changes, Artificial Intelligence (AI) based mechanisms are designed. Fuzzy logic has been used in [30] - [31], which makes decision according to distance, moving direction and speed of nodes. However, fuzzy logic is also dependent on thresholds and weights to be set in the rule base for making inferences. In [32], satellite images are used to detect buildings and obstacles to enable machine learning driven channel characterisation. The path with lowest propagation loss is used for message dissemination in [32]. RSU assisted deep learning based technique is developed for relay selection in [33]. It is pointed out that machine learning and deep learning algorithms require large processing power to handle huge amount of data and therefore they must require infrastructure support for implementation.

A security attack against availability is called jamming, in which an attacker causes interference to reduce SINR between sender and receiver to a level that an originated message is unable to reach its desired receiver and hence the information remains unavailable [10]. Therefore, selecting an appropriate relay node with least interference is desirable to ensure availability [27]. Availability aligns with the integrity protection attribute of 5G security requirements identified by 3GPP, i.e., a message should be protected from jammers and interferers to prevent degradation in SINR while it is being transmitted [34].

B. Privacy

Privacy means that only designated nodes in VANETs have permission to view other nodes’ original identities and control over granting access rights to nodes for viewing, sharing or originating messages [10]. The two attributes of privacy are as follows

- **Anonymity:** According to 3GPP requirement of privacy, anonymity is authenticating a node in a network without revealing its original identity [35]. One of the most commonly used approaches for anonymity in VANETs is public key cryptography [36], where public/private key pairs and digital certificates are used for pseudonymous communications. A third trusted party or CA is responsible for
issuance of key pairs and a digital certificate for authentication. Another approach is identity based and combined public key cryptography, which reduces the burden of CA as it uses private key for authentication instead of additional digital certificate by CA [37]. In both approaches, timely updates of key pairs are required for sustainable privacy preservation.

- **Confidentiality:** It implies that information contained in a message is secret and only a legitimate receiver should be able to access it. Eavesdropping attack attempts to break in confidentiality and retrieve information from messages without access rights [10]. PLS offers secrecy as a physical layer’s protection from eavesdropping. Secrecy rate of vehicular networks is analysed in [38] considering double Rayleigh fading environment. Only one legitimate receiver and one eavesdropper is assumed to be presented within the transmission range of sender, which is not very practical in high dense scenarios. In [39], the impact of eavesdropping is studied on various speeds and transmission ranges of nodes. It is concluded that secrecy rate can be maintained in presence of eavesdroppers at the cost of high transmission power and bandwidth. Cryptographic schemes are also used to protect confidentiality. As a trade-off they result in increased message size and additional computational and delay [39]. Confidentiality corresponds to ‘Access’ feature of 3GPP privacy requirement, i.e., only legitimate nodes are allowed to access a message [35].

### C. Trust

Trust is used to support security in VANETs. For example, trust ratings are used to evaluate message credibility. Also, an increase in trust or reputation score is served as an incentive to motivate nodes to cooperate and actively participate in message dissemination.

- **Incentive Distribution Mechanisms:** Incentive distribution mechanisms fall under the category of trust management models as they propose schemes to update trust ratings or reputation. However, there are two common types of incentive mechanisms: price-based and reputation based. In price-based strategy, messages are treated as commodities which are exchanged for virtual credits [40] - [41]. On the contrary, reputation-based strategies use measurement of trustworthiness to enforce cooperation. A threshold of reputation is set to distinguish malicious
behaviour from honest. A punishment scheme is applied for malicious behaviour [42] - [43]. Game theoretical analysis [44] is usually used to predict honest, selfish or malicious actions of nodes according to a designed incentive strategy. It is suggested that an integrated strategy of both price-based and reputation-based schemes is more effective in promoting cooperation and detecting malicious behaviour in MANETs [45]. A joint price and reputation based opportunistic routing solution for vehicular networks is presented in [46], where the originator pays a price to contributing nodes for successful delivery of message to a destination. The participating nodes are selected on the basis of their reputation. CA also receives a fraction of price for offering management services. A price-based approach in [47] claims to establish trust in VANETs by imposing price for initiating a message. The price is considered as a guarantee of message’s truthfulness.

Incentive mechanisms are not only effective to encourage selfish nodes to cooperate but also discourage cheating attempts by malicious nodes. Discouraging malicious and selfish behaviour is of crucial importance in VANETs, as they may hinder emergency message dissemination and risk road safety. A Payment Punishment scheme is proposed in [48] to demotivate malicious nodes to disseminate false messages.

Figure 2.3: Relationship between security, privacy, and trust in VANETs.
D. Decentralisation requirement for security, privacy and trust

Due to high dynamic and distributed nature of VANETs, decentralisation is an essential requirement for security, privacy and trust. VANETs must be able to independently make decisions, even when CA is out of reach. Since low latency is an important criterion when propagating messages to fast moving vehicles, a decentralised network in which nodes can select relay nodes among themselves independently without any CA is desired as a scalable and cost-effective approach [49]. Also, privacy and trust management would be compromised if CA is attacked or untrusted. Decentralization and third-party independence eliminate the possibility of single point failure in a network.

E. AI for security, privacy and trust

AI and machine learning algorithms are commonly integrated with a number of security and trust solutions, for example, misbehaviour detection by Support Vector Machine [50] and relay selection by deep learning [33]. There can be two approaches to perform machine learning or deep learning: on-device or off-device. In on-device learning, an inference model is trained on a node with its own private data without being shared with any other node [51]. It is considered as an effective solution for VANETs because of its privacy and low communication overhead, as it does not require data transfer to a cloud [52]. However, it is usually an uncooperative approach and an inefficient utilization of computation resource. It is because no knowledge of an on-device trained model is shared with other nodes and therefore, one node cannot utilize the benefits of a trained model produced by another. On the contrary, in a centralised solution or off-device learning, data from all connected nodes is gathered on a cloud or a central node, which produces inference model for all nodes [53]. Transmitting big data to a central node in VANETs is susceptible to propagation loss due to channel fading and is both bandwidth and time inefficient. Connectivity time among nodes is also limited. Furthermore, it is essential that the central node employs secure and privacy-preserving techniques for managing and storing big data.

- **Federated Learning (FL):** To manage the diverse nature of big data and also its privacy preserving requirements, FL is suggested as a suitable AI technique for various cooperative applications in vehicular communications [54]. In FL, nodes train local models individually on their own private data. Instead of whole data,
they share only local models with a central node or aggregator, which combines all local models to form a global model. Since FL relies on a central aggregator and VANETs require decentralization, it is challenging to implement conventional FL in VANETs. Also, an incentive mechanism is required to motivate nodes for contributing towards FL and discourage inaccurate local models. Furthermore, a malicious node can use false data or deliberately produce an inaccurate local model. Therefore, FL needs some means to detect malicious behaviour. It can be concluded that although FL is suitable for VANETs, it must be incorporated with solutions to offer decentralisation, security against malicious behaviour and a robust incentive distribution mechanism, all of which can be enabled by blockchain [55].

The next section of this chapter defines blockchain and its features and also discusses how blockchain can potentially be a solution to security, privacy and trust while offering decentralisation.

2.2 Blockchain

2.2.1 Introduction

A blockchain is a peer-to-peer electronic cash system in which transaction history is maintained in the form of immutable timestamped blocks. As shown in Figure 2.4, each block in a blockchain contains a cryptographic link to the previous block known as hash [56]. Blockchain based cryptocurrency, bitcoin, was first introduced by Satoshi Nakamoto in 2008 [5]. Since then, multiple variations of blockchain and cryptocurrencies have been released.

![Figure 2.4: The blockchain structure.](image)
A conventional blockchain inherently possesses a network of nodes. Whenever a transaction request is made, it is broadcasted across the network. Nodes have the right to validate a transaction and mine the validated transaction into a block containing an encrypted hash of previous block, thereby making it cryptographically secure. Nodes which append a block to the blockchain after validation and broadcast it in the network are known as miners. Miners have to go through a mutual agreement called consensus algorithm to create a block which makes it fraud-proof [57]. For each mined block, a miner earns reward in relevant cryptocurrency, which is usually paid by the proposer of a validated transaction [58]. All nodes are responsible to update their copy of blockchains regularly to ensure consistency in the entire distributed ledger [59].

### 2.2.2 Features

A blockchain-enabled network offers the following features

- **Decentralisation**: The methodology of validating and recording transactions in a blockchain is decentralised and independent of CA. This feature makes it suitable for distributed networks, for example in VANETs [60] - [61]. Also, blockchain is not stored in a central server or node. All nodes possess a copy of blockchain. If a malicious node attempts to modify contents of a block, it must do it in every copy, which can be difficult [62] - [63].

- **Credibility**: Nodes go through a consensus algorithm to validate a transaction in a block. If a malicious node attempts to propose a fake transaction, it cannot be included in a block without being validated by a consensus algorithm [64] - [65]. The consensus algorithm can be used to validate a message instead of a financial transaction in VANET thus ensuring credibility of a message.

- **Privacy**: The transactions are made in blockchain using private keys. A key can also be changed for different transactions. This feature introduces anonymity in the network [66]. The blocks are also encrypted and can be accessed with correct pair of keys, ensuring confidentiality of data [67]. Similarly, messages or transactions of incentives in VANET can also be anonymously recorded and kept confidential in blockchain.
• **Automation**: Blockchain can enforce a set of rules and allow automation of multi-step processes and interaction among nodes using smart contract. Smart contracts are self-executing scripts residing on the blockchain, which run independently in a prescribed manner [68]. Nodes can store a set of agreed rules to validate a transaction in smart contract. It executes on receiving transaction proposals and automatically reaches consensuses in less time without the need of third party [69]. Recent works have utilised smart contract to automate secure charging of electric vehicles [70] - [71].

### 2.2.3 Consensus Algorithms

The consensus algorithm of a blockchain is designed to solve Byzantine general’s problem [72] - [73]. This problem considers the presence of faulty or malicious nodes in a network. These nodes can behave differently to other nodes. For examples, they may ignore a valid transaction or propose a fake transaction [74]. A consensus algorithm works in two ways, i.e., it either elects a miner in a network which validates a block through a competition or eligibility criteria, or it defines a set of rules run by smart contract to validate a transaction. The consensus is finalised in a distributed manner and is intended to work without errors in the presence of faulty nodes. The performance of a consensus algorithm is evaluated by the following metrics

- **Tolerance**: Number of malicious nodes it can control without altering the original validity status of a transaction and its ability to resolve forks and prevent cheating.
- **Latency**: Time required to validate and propagate a transaction.
- **Throughput**: Number of blocks generated per second.

Some of the common consensus algorithms are:

- **Proof-of-Work (PoW)**: In PoW, the nodes solve a complex cryptographic challenge which typically takes around 10 minutes to complete. The node which solves the challenge first is elected as miner. PoW can tolerate around 50% malicious nodes in a network [75]. However, due to its complexity and huge time consumption, it is
not considered appropriate for mobile networks with short-lived connectivity span, such as, VANETs [76].

- **Proof-of-Stake (PoS):** It is an alternative to PoW which can save computational energy and time. It works under the assumption that the node with the highest number of stakes is less likely to be faulty [77]. The nodes prove the amount of credit or stakes owned. The node with largest number of stakes becomes miner. It takes considerably less time than PoW. However, it is unfair to nodes owning less stakes and therefore promotes oligarchy in the network. Only nodes with high stakes are able to be active in the network, which demotivates other nodes to participate. Therefore, it is not suitable to applications where positive cooperation is required from every node, for example, sharing resources, information exchange and emergency message dissemination.

- **Proof-of-Elapsed-Time (PoET):** In PoET, each node generates a random number, following a probability distribution, to determine how long it needs to wait before generating a block. The node with shortest waiting time is elected as miner. Generating a random number requires significantly less energy as compared to solving a cryptographic challenge. However, it is proved to be less secure in theory because malicious nodes can easily generate the shortest time to become miner [78].

- **Practical Byzantine Fault Tolerant (PBFT) algorithm:** It is a voting-based consensus algorithm aimed to achieve high throughput and low computation cost. As shown in Figure 2.5, it consists of the following stages [79] - [80]:
  
  - An originator initiates a transaction proposal and requests nodes to vote for validation.
  - Nodes, also known as endorsers, vote if the transaction is valid.
  - If the transaction is validated by a threshold number of votes, it is added as a block into blockchain.
A comparison of the above discussed consensus algorithms is presented in Table 2.1.

<table>
<thead>
<tr>
<th>Consensus</th>
<th>Energy saving</th>
<th>Time saving</th>
<th>Discriminatory</th>
<th>Secure</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoW [5]</td>
<td>No</td>
<td>No</td>
<td>Yes (towards computational power)</td>
<td>Yes</td>
</tr>
<tr>
<td>PoS [77]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (towards amount owned)</td>
<td>Yes</td>
</tr>
<tr>
<td>PoET [78]</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PBFT [79] - [80]</td>
<td>Yes</td>
<td>Partial</td>
<td>No</td>
<td>Partial</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of consensus algorithms.

### 2.2.4 Blockchain Variations

Due to increasing applications of blockchain outside finance, many variations in its framework and structure are now proposed and widely practised. Initially, a blockchain was designed to store transactions of cryptocurrency. When cryptocurrency is managed by blockchain, each block contains a set of verified transactions and a cryptographic hash linked with previous block, as shown in Figure 2.4: The blockchain structure. However, a block can be used to store a verified record of any item according to required
application, for example, update in reputation rating of a node or message exchanged at each hop, as shown in Figure 2.6.

![Figure 2.6: Variations in block contents.](image)

**A. Permissioned and Permissionless Blockchains**

Blockchains were conventionally designed to be permissionless. Nowadays, blockchains are broadly classified into the following two categories [81]:

- **Permissionless blockchain**: In permissionless blockchain, all nodes have equal rights to access and generate blocks, thus ensuring transparency in the network. PoW and PoS are mostly used with permissionless blockchain [56].
- **Permissioned blockchain**: In a permissioned blockchain, specific roles and access limitation can be associated to certain nodes. Permissioned blockchain usually employs limited number of nodes in a consensus. PBFT algorithm is common for achieving consensus among a group of nodes in permissioned blockchain [82]. Considering different roles in ITS, such as automotive manufacturers, government
and transportation authorities, a permissioned blockchain is considered appropriate for vehicular applications [16].

B. Hierarchical Blockchains

Variations are also present in distributed ledger of blockchain to ensure high throughput. A conventional blockchain ledger is horizontal, where each block is only linked with the previous block, as shown in Figure 2.4. However, several hierarchical structures of blockchain have now been proposed to improve throughput and latency of block generation. Such solutions contain parallel tiers of off-chain blocks, all connected to a main block, as shown in Figure 2.7. Miner is usually considered as a leader in hierarchical blockchain. The subordinates of miner are used to validate transactions at the same time and generate blocks containing the hash of a previous block [83]. In this way, multiple blocks containing one hash are added in parallel.

In a conventional (non-hierarchical) blockchain, parallel blockchain extension is called fork, as shown in Figure 2.8. As a common practice, blockchain picks one of the parallel blocks to continue, and meanwhile, disqualifies other forking blocks by longest chain acceptance protocol [84]. Forks also lead to creation of malicious chains [75]. On the contrary, in a hierarchical structure, parallel blocks are not considered as forks. In a hierarchical blockchain, blocks are classified into two types: keyblock and microblock. Instead of a linear ledger, microblocks representing off-chain transactions are added in parallel, whereas keyblocks are main blocks which are appended horizontally in the blockchain by a leader or a central node, for example, RSU in vehicular networks. Practical implementation of a conventional blockchain in VANETs is challenging. Due to limited connectivity duration in V2V communications, moving nodes may not always have an updated blockchain ledger, which leads to a fork situation, as shown in Figure 2.9 (a). To address this issue, the hierarchical structure of blockchain is proposed for vehicles [85]. As shown in Figure 2.9 (b), parallel addition of microblocks does not disturb the main linear ledger and forks are not disqualified but accepted as off-chain micro-transactions recorded in a decentralised manner.
Figure 2.7: A hierarchical blockchain structure.

Figure 2.8: Fork occurrence in conventional blockchain.
2.2.5 Blockchain for Security, Privacy and Trust

The prospect of blockchain as a security, privacy and trust solution has drawn increasing interests in communications research community. This subsection reviews some of the existing approaches of blockchain implementation in ad-hoc networks, particularly in VANETs. Table 2.2 summarises the advantages and challenges of discussed applications.
Table 2.2: Advantages and challenges of blockchain applications.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Applications</th>
<th>Advantages</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Credibility Evaluation [86] - [87]</td>
<td>Decentralisation</td>
<td>Choice of suitable consensus is necessary for low latency</td>
</tr>
<tr>
<td></td>
<td>Certificate / Key management [88] - [89]</td>
<td>Automation</td>
<td>Often needs CA and complete decentralisation is difficult to achieve</td>
</tr>
<tr>
<td></td>
<td>Confidentiality Protection [90]</td>
<td>Private blockchain with access control</td>
<td>Possibility of eavesdropping</td>
</tr>
<tr>
<td>Privacy</td>
<td>Reputation Management [7], [91] - [92]</td>
<td>Decentralisation and immutability</td>
<td>Difficult to find suitable consensus: PoW is time consuming, PoS does not motivate nodes with low reputation</td>
</tr>
<tr>
<td></td>
<td>Incentive Mechanism [37], [93] - [94]</td>
<td>Privacy and immutability</td>
<td>Additional cryptographic computations</td>
</tr>
</tbody>
</table>

A. **Security**

- **Credibility:** The distributed ledger of blockchain storing trust ratings or consensus to reach a mutual agreement provides solutions to assess message credibility. Trust or reputation ratings based blockchain solutions are commonly used as decentralised pre-event approaches to evaluate message credibility in VANETs. A permissioned blockchain of reputation using PoW consensus is proposed in [86], which is used to validate messages on the basis of sender's reputation. In [95], smart contract is utilised to update trust ratings of nodes, which are then used to classify honest or malicious message originator. Blockchain can also be used to validate a message by post-event approach. A blockchain-based solution can achieve consensus if satisfactory endorsements are received as a proof-of-event [96]. PBFT
based voting is used in [37] and [87] to collect endorsements for message credibility. Also, as a hybrid approach in [7], Delegated PoS consensus only allows selected nodes take part in post-event validation. These nodes are selected on the basis of their trust ratings.

- **Availability:** To ensure availability of message in VANETs, a relay selection Stackelberg game model considering trust ratings of nodes stored in blockchain is proposed in [97]. However, it is worth noting that formulating a blockchain consensus exclusively for enabling distributed relay selection is a novel contribution of this thesis.

**B. Privacy**

- **Anonymity:** As discussed in Section 2.1, public key cryptography is used to ensure anonymity in VANETs. In large scale networks, it may result in running several cryptographic operation requests at the same time which is not a feasible solution [98]. Blockchain based solutions are proposed in literature for third party independence and dynamic key management [88]- [99]. A blockchain managed by CA to assign or revoke encrypted keys in VANET is proposed in [88]. However, when a CA is responsible for issuing or revoking certificates for private vehicular communications, there is a risk of single point of failure affecting the whole system. Therefore, distributed ledger of blockchain is recommended in [100], where multiple nodes keep the record of issued or revoked certificates. A similar approach is presented in [101], where blockchain is used for key management and RSUs are assigned as validators. Specific nodes or validators perform the role of CA in [100]- [101]. Therefore, they can be considered as partially decentralised solutions. Key management through blockchain as a potential research area for future has also been discussed in [89].

- **Confidentiality:** The confidentiality of VANETs can be protected by implementing encrypted blocks and access control schemes. However, there still remains a possibility of eavesdropping even when blockchain is used because of the broadcasting nature of both VANETs and block announcement procedure [102]. In [90], the nodes are allowed to register into a private blockchain only when the secrecy rate of physical layer exceeds a certain threshold. A blockchain based
federated learning mechanism is proposed in [103]. Despite the privacy preserving nature of both federated learning and blockchain, an additional technique is applied to protect location confidentiality of nodes.

C. Trust

- **Reputation Management**: The distributed ledger of blockchain is commonly utilised as a decentralised storage medium for recording trust ratings or reputation scores [91] - [104]. In [7], a blockchain based reputation management solution is proposed with PoS consensus, where reputation is considered as the stake of nodes. The nodes report their reputation opinions of other nodes to a nearby RSU. To carry out consensus protocol, the reputation values are downloaded from RSU and a mining node is elected on the basis of highest reputation. A node is considered malicious if its reputation value falls below a certain threshold. A similar approach is presented in [92], where RSUs store reputation of nodes and maintain blockchain. To carry out the consensus, mining RSUs undergo a joint PoW and PoS algorithm, where reputation is regarded as a stake. Both mechanisms rely on RSUs for blockchain management.

- **Incentive Distribution Mechanisms**: Blockchain also manages distribution of incentives by recording them in the form of transactions involving cryptocurrency or virtual credit. Secure blockchain-based incentive mechanisms are proposed in the literature to encourage cooperative message delivery and data sharing in distributed peer-to-peer (P2P) applications. In [93], data storage among nodes in wireless sensor networks is incentivised and proof of stored data is proposed as the consensus algorithm. An example of incentivized caching for content delivery based on blockchain is presented in [105]. In [106], an incentive strategy is proposed for crowdsensing applications where privacy is preserved by cryptographic signatures. In [107], a pricing strategy to ensure successful message delivery using blockchain is presented and proved to be secure against the collusion of intermediate nodes and receiver using the game theory. Similarly, in [108], P2P data sharing using permissionless blockchain is proposed. In both [107] and [108], the incentive mechanism is proved to encourage cooperation among nodes by including a mandatory charge paid by transmitting nodes. Incentive-based message
relaying in distributed P2P applications using the blockchain is also proposed in [109] and proved to be secure against selfish behaviour. In VANETs, a blockchain based incentive mechanism which uses virtual credits, known as CreditCoin, as a reward for forwarding the message is proposed in [37]. The transactions of CreditCoins are managed by blockchain, which is independent to V2V communications. The message dissemination solution is not completely decentralised because it relies on RSUs or official public vehicles as servers to conduct consensus. A decentralised solution of incentive distribution through blockchain is proposed in [61]. It is suggested to automatically punish negative behaviour in an ITS. For example, behaviours of over-speeding and ignoring traffic lights are punished by imposing fine, recalculating tax or insurance, which can be implemented by the use of smart contract without requiring a CA. In [94], the incentive-based message delivery in wireless ad hoc networks for smart cities and ITS is presented, where the message is validated using PBFT, and the incentives and privacy are controlled using blockchain.

2.3 Summary

This chapter first introduces VANET and discusses requirements and approaches for security, privacy and trust for message dissemination. Secondly, blockchain, its consensus algorithms, variations and promising solutions to security, privacy and trust are reviewed. It can be concluded that the features of blockchain can offer solutions to many challenges in VANETs. Specifically for message dissemination, blockchain is a potential approach to provide decentralisation, security and trust. Table 2.3 summarises challenges associated with message dissemination in VANET and corresponding solutions provided by the blockchain.

Some of the existing approaches to implement blockchain in VANETs are discussed in this chapter. However, it is crucially important that technical challenges, such as dynamic nature of VANETs, short-lived connectivity, broadcast storm, packet collision, and computing complexity are addressed while implementing the blockchain. Therefore, current blockchain solutions may not be fully compatible with vehicular networks in practice and new approaches are in need for VANETs. It is well-acknowledged that
security, privacy and trust solutions are widely discussed in literature, but for a message dissemination solution in VANETs, there is a need to establish an integrated approach in which efficient and trusted communications in a vehicular network can be managed in a decentralised fashion. Also, a detailed performance analysis is required to evaluate the practical feasibility of blockchain in VANETs.

<table>
<thead>
<tr>
<th>Issues in VANET</th>
<th>Blockchain based solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message validation</td>
<td>Consensus algorithm</td>
</tr>
<tr>
<td>Trust without third party dependence</td>
<td>Decentralised ledger</td>
</tr>
<tr>
<td>Privacy requirement</td>
<td>Cryptographic hashes</td>
</tr>
<tr>
<td>Broadcasting storm / relay selection</td>
<td>Miner election by consensus</td>
</tr>
<tr>
<td>Incentive distribution</td>
<td>Miner incentives</td>
</tr>
</tbody>
</table>

Table 2.3: VANET issues and opportunities using blockchain.
Chapter 3 – Voting based Blockchain for Message Validation and Relay Selection

Some existing blockchain consensus and their limitations are discussed in Chapter 2. This chapter introduces a novel voting-based Proof-of-Quality-Factor (PoQF) consensus as a solution appropriate to VANETs. PoQF incorporates message validation and a decentralised relay selection mechanism. The motivation and challenges in conventional voting based PBFT are discussed in section 3.1. The rest of the chapter consists of three main sections. In section 3.2, we explain the formulated model and approach, and propose two ways of incentive distribution mechanisms for message validation and multi-hop relaying. In section 3.3, we theoretically analyse the proposed solution. The simulation results of the proposed solution are discussed in section 3.4.

3.1 Overview

3.1.1 Motivation

As discussed in previous chapters, blockchain applications in vehicular networks can offer many advantages including decentralisation and improved security. However, one of the challenges to implement blockchain in VANETs is to select an appropriate consensus algorithm particularly satisfying vehicular application requirements. The large latency and high computation costs of PoW and unfairness of PoS are already discussed. If PoW is employed in VANETs, it must be run by RSU [110] or computation offloading to edge devices is required [111]. PoS, the alternative to PoW, usually finds its applications in reputation management [7], [92]. Another consensus algorithm, PoET, offers the fairness with less computation workload but its weakness in security and vulnerability in the presence of malicious nodes is proved in literature [78]. Selection of consensus for blockchain applications in VANETs is widely discussed in literature and the voting based PBFT is one the most recommended approaches for connected vehicles [37], [94], [112]-[113], which offers high throughput and has the ability to negotiate message validity [94], [112].
As shown in Figure 3.1, the voting-based consensus in VANETs is basically analogous to threshold-based message validation in which a message is considered valid only if it is confirmed by a threshold number of nodes located in a close proximity of an originator (\textit{ORG}) [37]. Based on existing literature, Table 3.1 summarises consensus algorithms used...
for VANETs and indicates several usages of PBFT including message validation by voting, high throughput and ability to finalize transactions independently without RSU reliance.

3.1.2 Contributions

a) Threshold number of votes

In a voting-based consensus, the threshold value to finalise a validation is crucial. A low threshold value may lead to false validation, whereas a high threshold value can result in an increased latency. This chapter identifies a range of threshold values with varying percentage of malicious nodes which result in a decreased percentage of failure in validating a message by the proposed PoQF. The detailed derivations are presented in Proposition 3.4 and Assuming that

\[ p_m^2 \mu_m + \mu_m \approx (1-p_m)^2 \left( 1-e^{-\frac{\mu_m^2}{2\mu_h}} \right) \approx 0, \quad (3.29) \]

leads towards the condition, \( p_m > 0.5 \). It shows that the condition in (3.28) is fulfilled when \( p_m > 0.5 \) and therefore, according to (3.26), minimum \( FV_{UB} \) is obtained when \( n_{th} > \max(\mu_m, \mu_h) \). Since \( \mu_m > \mu_h \) only when \( p_m > 0.5 \), it means that minimum \( FV_{UB} \) at \( p_m > 0.5 \) is attained when \( n_{th} > \mu_m \).

Proposition 3.5.

b) Security against malicious nodes

In a conventional PBFT, the nodes broadcast their votes only for a true transaction. The transaction is automatically marked as false if the threshold number of votes are not received. As a novel contribution, this chapter proposes a consensus algorithm which receives both true and false votes. Certain rules described in section 3.2.2 decide message validity if threshold number of votes are not received in either case or if there is a tie. These features enhance the security of the proposed PoQF.

c) Relay node selection

Traditionally, voting based consensus is used to verify credibility of a message. This chapter incorporates a relay selection mechanism in voting. Along with the votes to validate message, each node also computes and broadcasts its score, i.e., Quality Factor to become a relay. Similar to the miner election in a blockchain consensus, the node with the highest Quality Factor is elected as relay for message dissemination.
3.2 System Modelling and the Proposed Blockchain Design

3.2.1 Components

The proposed solution consists of following components:

- **Originator (ORG):** It is the node which is involved in an incident. ORG is the sender $s$ of the message at first hop, i.e., when $n_{hop} = 1$, where $n_{hop}$ is the hop number.

- **Mining node:** A node $i$ which receives and responds to the message from sender $s$ by broadcasting its vote for message validity and Quality Factor ($QF_i$) to become a relay node.

- **Quality Factor of node $i$ ($QF_i$):** It is the parameter on the basis of which a node $i$ qualifies to become a relay node. The calculations to compute $QF_i$ are explained later in this section.

- **Relay node (RLY):** A node $i$ with highest $QF_i$ becomes RLY at a particular $n_{hop}$.

- **Proof-of-Quality-Factor (PoQF):** It is the voting and $QF_i$ based consensus algorithm carried out in a decentralised manner by nodes in VANET to validate and disseminate a message via blockchain.

- **Microblock:** It is the block generated by mining node $i$ recording its $QF_i$ and vote towards validity of a message. Microblocks are appended in parallel in a blockchain at each hop.

- **Keyblock:** It is the block generated by RLY and records final validity status of a message after voting.

- **Waiting Time ($\tau_i$):** It is a random waiting time for which node $i$ waits before generating a microblock.

- **Failure in Validation (FV):** It is the probability that the original validity of a message is inverted after PoQF consensus at $n_{hop} = 1$.

- **Transaction Charge (TC):** $TC > 0$ is paid by mining node $i$ while voting. The motivation behind introducing $TC$ is to discourage malicious false votes.

- **Call Compensation (CC):** As a compensation of causing an incident, CC is the amount paid by ORG. It is distributed as an incentive to cooperative nodes, i.e., honest mining nodes and RLY at each hop.
3.2.2 The Proposed PoQF Consensus

The proposed PoQF consists of four stages, as illustrated in Figure 3.2. At the first stage, an incident occurs and a message is initiated by ORG involved in the incident. The message is analogous to a transaction proposal in a consensus that requires validation.

At the second stage, each mining node \( i \) generates a microblock, in which it records its vote towards validity of the message and \( QF_i \), to become a potential RLY. A node \( i \) waits for time \( \tau_i \) before it announces a microblock. \( \tau_i \) is a randomly generated number following uniform distribution, i.e., \( \tau_i \sim \mathcal{U}(a_{\tau_i}, b_{\tau_i}) \), where \( a_{\tau_i} \) and \( b_{\tau_i} \) are lower and upper limits of \( \tau_i \), respectively, which are dependent on \( QF_i \). The motivation behind using \( \tau_i \) is three-fold: one is to prevent all nodes from transmitting at the same time and causing packet collision, second is to introduce fairness by giving less waiting time to nodes with higher \( QF_i \) and the third is to ensure randomization if node \( i \) and node \( j \) have \( QF_i = QF_j \). Using uniform distribution to randomize scheduling of messages so as to avoid packet collision has been previously used in literature [114].

![Figure 3.2: The proposed PoQF consensus.](image-url)
At the third stage, node $i$ is selected as a $RLY$ if it fulfills two conditions. First, it has received at least $n_{th}$ microblocks with the same votes as its own. Second, its $QF_i$ is the highest among all microblocks with the same votes as its own. The motivation behind these two conditions instead of $QF_i$ only is to enhance security of PoQF. For example, if a malicious node $i$ with the highest $QF_i$ among all mining nodes votes false for an originally true message and receives $n_{th}$ microblocks with true votes, it cannot become $RLY$ and earn incentive. Similarly, if an honest node $i$ with the highest $QF_i$ votes false for an originally false message but receives $n_{th}$ microblocks with true votes, it cannot become a $RLY$ to forward a false message. $RLY$ will forward the message only if it is validated as true but always generate a keyblock to record message validity after PoQF and transactions, which are related to incentive distribution. As shown in Figure 3.3, if no node receives at least $n_{th}$ microblocks with the same votes as its own until 1s, i.e. the maximum allowable
latency for emergency message dissemination [3], the message is considered as false and a keyblock to record such transaction will be generated by the mining node \(i\) with highest \(QF_i\), which voted false. If two \(RLYs\) with opposing votes are selected (one with true vote and another with false vote), the message is considered true so that the cooperation may not be stopped in case of a true incident. The value of \(n_{th}\) corresponding to real traffic conditions is communicated to nodes by RSU.

The fourth stage is continuation of message dissemination. If the message is validated as true, it is disseminated after a new \(RLY\) selection by PoQF at each hop until it reaches a maximum specified distance or until \(n_{\text{hop}} \leq n_{\text{hop}}^{\text{max}}\), where \(n_{\text{hop}}^{\text{max}}\) is the maximum number of hops up to which a message is required to be forwarded. It is noted that votes to validate a message are not required for \(n_{\text{hop}} > 1\). It is simply because the validation of message has been done by adjacent witness nodes (mining nodes at \(n_{\text{hop}} > 1\)) through a camera or location/speed verification [16]. All other nodes beyond the first hop may not have access to validate the \(ORG\).

### 3.2.3 \(QF_i\) Calculations

\(QF_i\) determines the quality of mining node \(i\) at the time when it forwards the message as \(RLY\). Each node regularly shares its position and velocity via beacon message. As shown in Figure 3.2, two consecutive beacons messages are exchanged at \(t_0\) and \(t_1\) before the occurrence of incident. To compute \(QF_i\), node \(i\) makes probability based predictions of distances with its neighbour nodes at time \(t_2 = t_{\text{icd}} + \bar{T}_{\text{delay}}\), where \(t_{\text{icd}}\) is the time at which the incident message is received from the sender \(s\) (\(ORG\) or previous \(RLY\)) and \(\bar{T}_{\text{delay}}\) is the mean time delay to finalize consensus and is described in details in Section 3.4.3.3.2. As \(QF_i\) decides the \(RLY\), it is governed by two factors [115]: the probability of success that a node's transmission can reach to all of its neighbour nodes, i.e., Quality of SINR at \(t_2\), \(Q(SINR_i^{t_2})\), and the probability that its distance to the sender \(s\) is larger than a threshold for ensuring successful transmission over longer distances, i.e., Distance Factor at \(t_2\), \(DF_i^{t_2}\). Hence, \(QF_i = Q(SINR_i^{t_2}) \cdot DF_i^{t_2}\).
a) \( Q(SINR_{t_2}^t) \)

If mining node \( i \) becomes RLY, the SINR of a signal received at receiving node \( j \) from node \( i \) at \( t_2 \) is

\[
SINR_{i,j} = \frac{(d_{i,j})^{-\alpha}}{P_{\text{noise}} + \sum_{k=1,k\neq i}^{n_{itf}} (d_{j,k})^{-\alpha}}, \tag{3.1}
\]

where \( \alpha \) is the path loss exponent and its value depends on fading environment [27], \( d_{i,j} \) is the distance between node \( i \) and node \( j \), \( d_{j,k} \) is the distance between node \( j \) and interfering node \( k \), \( n_{itf} \) is the number of interference nodes and \( P_{\text{noise}} \) is the noise power.

For a successful message transmission, it is required that the SINR exceeds a certain threshold \( \beta_1 \), i.e., \( SINR_{i,j} \geq \beta_1 \). The probability that \( SINR_{i,j} \geq \beta_1 \) at \( t_2 \), i.e., \( Pr(SINR_{i,j}^t \geq \beta_1) \) is given as

\[
Pr(SINR_{i,j}^t \geq \beta_1) = Pr\left( \frac{(d_{i,j}^t)^{-\alpha}}{P_{\text{noise}} + \sum_{k=1,k\neq i}^{n_{itf}} (d_{j,k}^t)^{-\alpha}} \geq \beta_1 \right) \tag{3.2}
\]

\[
= Pr\left( d_{i,j}^t \leq \left( \beta_1 \left( P_{\text{noise}} + \sum_{k=1,k\neq i}^{n_{itf}} (d_{j,k}^t)^{-\alpha} \right) \right)^{-\frac{1}{\alpha}} \right),
\]

where \( d_{i,j}^t = d_{i,j}^{t_1} + \Delta d_{i,j}^{\Delta t} \) is the distance between node \( i \) and node \( j \) at \( t_2 \) and \( \Delta d_{i,j}^{\Delta t} \) is the relative distance change between node \( i \) and node \( j \) during \( \Delta t = t_2 - t_1 \). \( d_{i,j}^{t_1} \) can be obtained from the beacon message received at \( t_1 \) and the expected value of \( \Delta d_{i,j}^{\Delta t} \) can be found using Probability Density Function (PDF) of standard Gaussian distribution.

Referring to the results in [27] and [116], the velocity of a node \( i \) follows a standard Gaussian distribution, i.e., \( v_i \sim \mathcal{N}(0, \sigma_i^2 t) \), where \( \sigma_i^2 = \frac{(v_i^{t_1} - v_i^{t_0})^2}{t_1 - t_0} \) is variance of \( v_i \) and \( v_i^{t_0} \) and \( v_i^{t_1} \) denote \( v_i \) at \( t_0 \) and \( t_1 \), respectively, which are shared by node \( i \) via beacon messages. \( \Delta d_{i,j}^{\Delta t} \) is defined as

\[
\Delta d_{i,j}^{\Delta t} = (v_i^{t_1} - v_j^{t_1} + \Delta v_i^{\Delta t} - \Delta v_j^{\Delta t}) \Delta t, \tag{3.3}
\]
where $\Delta v_i^{\Delta t}$ is the change in $v_i$ during $\Delta t$. By the principle of linear combination of Gaussian variables, $\Delta v_i^{\Delta t} \sim \mathcal{N}(0, \sigma_i^2 \Delta t)$, $\Delta v_i^{\Delta t} - \Delta v_j^{\Delta t} \sim \mathcal{N}(0, (\sigma_i^2 + \sigma_j^2) \Delta t^3)$ and hence, $\Delta d_{i,j}^{\Delta t} \sim \mathcal{N}(0, \sigma_i^2 \Delta t)$.

If $v_i^{t_2}$ is not known, (3.2) can be calculated by assuming $\Delta d_{i,j}^{\Delta t}$ as a standard Gaussian variable.

Each node $i$ calculates (3.2) as per Theorem 3.1. with respect to all its neighbour nodes. As $n_{ltf}$ is the number of neighbours of node $j$ except node $i$, $n_{ltf}$ and $d_{j,k}^{t_2}$ are unknown to node $i$. It can estimate the expected values to find $Pr(SINR_{i,j}^{t_2} \geq \beta_1)$. Hence

$$\left( \beta_1 \left( P_{\text{noise}} + \sum_{k=1}^{n_{ltf}} E(d_{j,k}^{t_2})^{\alpha} \right) \right)^{-\frac{1}{\alpha}}$$

in (3.2) can be rewritten as

$$\left( \beta_1 \left( P_{\text{noise}} + E(n_{ltf}) E(d_{j,k}^{t_2})^{\alpha} \right) \right)^{-\frac{1}{\alpha}},$$

where $E(.)$ denotes expected value. The location of nodes (vehicles) on road is assumed to follow an independent homogeneous spatial Poisson distribution with density parameter $\lambda_V$ nodes/m$^2$ on a two-dimensional road segment with no separation of lanes in order to make it general and allow dynamic movement of nodes [117]. Therefore, $E(n_{ltf})$ can be estimated as the number of nodes within the transmission range of node $j$. Assuming that transmission range of each node is a uniform circular area with radius $R$, $E(n_{ltf})$ can be calculated as the number of nodes inside the area excluding node $i$, i.e.,

$$E(n_{ltf}) = \sum_{k=1}^{\pi R^2 \lambda_V} \left( k \frac{(\pi R^2 \lambda_V)^k}{k!} e^{-\pi R^2 \lambda_V} \right) - 1,$$

where $\lambda_V$ corresponding to real traffic conditions is communicated to nodes via RSU. It is noted that an adaptive $\lambda_V$ corresponding to real traffic conditions is out of the scope of this chapter but can be locally estimated by calculating the number of received beacons [118] or with the use of edge computing servers [119].

- **Lemma 3.1:** $E(d_{j,k}^{t_2}) = \frac{2}{3R^2} \left( R^3 - d_{\text{neigh}}^{min} \right)^3$, where $d_{\text{neigh}}^{min}$ is the minimum allowed distance between two neighbour nodes.

**Proof:** The PDF of interference nodes at location $(X,Y)$ within the area $\pi R^2$ is defined in [120] as $1/\pi R^2$. Therefore, $E(d_{j,k}^{t_2})$ can be calculated as
Assuming transmission range is a uniform circle with radius $R$, then $X = R\cos \phi$ and $Y = R\sin \phi$. Therefore, (3.4) becomes

$$E(d_{j,k}^{t_2}) = \int_{a_{\text{neigh}}^{\min}}^{R} \int_{0}^{2\pi} \frac{z^2}{\pi R^2} \text{d}\phi \text{d}r$$

$$E(d_{j,k}^{t_2}) = \frac{2}{3R^2}(R^3 - a_{\text{neigh}}^{\min})^3.)$$

• **Theorem 3.1:** $\Pr(\text{SINR}_{i,j}^{t_2} \geq \beta_1) =

$$\begin{cases}
\frac{1}{2} \left( \text{erf} \left( \frac{\Delta d_{i,j}^{\Delta t}}{\sqrt{2(\sigma_i^2 + \sigma_j^2)\Delta t^3}} \right) - \text{erf} \left( \frac{-\Delta d_{i,j}^{\Delta t}}{\sqrt{2(\sigma_i^2 + \sigma_j^2)\Delta t^3}} \right) \right), & d_{i,j}^{t_1} \leq d_x \\
1 - \frac{1}{2} \left( \text{erf} \left( \frac{\Delta d_{i,j}^{\Delta t}}{\sqrt{2(\sigma_i^2 + \sigma_j^2)\Delta t^3}} \right) - \text{erf} \left( \frac{-\Delta d_{i,j}^{\Delta t}}{\sqrt{2(\sigma_i^2 + \sigma_j^2)\Delta t^3}} \right) \right), & \text{otherwise,}
\end{cases}$$

where $d_x = \left(\beta_1 \left( P_{\text{noise}} + E(n_{\Delta t})E(d_{j,k}^{t_2})^{-\alpha} \right) \right)^{\frac{1}{\alpha}}$ and $E(d_{j,k}^{t_2})$ is defined in Lemma 3.1.

**Proof:** Since $d_{i,j}^{t_2} = d_{i,j}^{t_1} + \Delta d_{i,j}^{\Delta t}$, it is essential to first find the probability of $\Delta d_{i,j}^{\Delta t} \leq d_x - d_{i,j}^{t_1}$. If $d_{i,j}^{t_1} \leq d_x$, the actual required communication distance, $\Delta d_{i,j}^{\Delta t}$ can be calculated as in [116]

$$\Delta d_{i,j}^{\Delta t} = \begin{cases}
\left( d_x + d_{i,j}^{t_1}, & \text{Case 1,} \\
\left( d_x - d_{i,j}^{t_1}, & \text{Case 2,}
\end{cases}$$

where Case 1 is either of the following

- node $i$ and node $j$ are moving towards each other
- node $i$ is in front of node $j$, both moving in same direction, and $v_i < v_j$
- node $j$ is in front of node $i$, both moving in same direction, and $v_i > v_j$
and Case 2 is either of the following

- node \( i \) and node \( j \) are moving away from each other
- node \( i \) is in front of node \( j \), both moving in same direction, and \( v_i > v_j \)
- node \( j \) is in front of node \( i \), both moving in same direction, and \( v_i < v_j \).

Therefore, PDF of \( \Delta d_{i,j}^{\Delta t} \) can be defined as

\[
f(\Delta d_{i,j}^{\Delta t}) = \frac{1}{\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)\Delta t^3}} e^{-\frac{(\Delta d_{i,j}^{\Delta t})^2}{2(\sigma_i^2 + \sigma_j^2)\Delta t^3}}.
\] (3.7)

Consider both acceleration and deceleration, the Cumulative Distribution Function (CDF) can be calculated as

\[
F(\Delta d_{i,j}^{\Delta t}) = \int_{-\Delta d_{i,j}^{\Delta t}}^{\Delta d_{i,j}^{\Delta t}} f(\Delta d_{i,j}^{\Delta t}) d(\Delta d_{i,j}^{\Delta t}).
\] (3.8)

As \( F(\Delta d_{i,j}^{\Delta t}) = Pr(d_{i,j}^{t_2} \leq d_x) = Pr(SINR_{i,j}^{t_2} \geq \beta_1) \),

\[
Pr(SINR_{i,j}^{t_2} \geq \beta_1) = \frac{1}{2} \left( \text{erf} \left( \frac{\Delta d_{i,j}^{\Delta t}}{2(\sigma_i^2 + \sigma_j^2)\Delta t^3} \right) - \text{erf} \left( \frac{-\Delta d_{i,j}^{\Delta t}}{2(\sigma_i^2 + \sigma_j^2)\Delta t^3} \right) \right).
\] (3.9)

Otherwise, if \( d_{i,j}^{t_1} > d_x \), the actual required communication distance \( \Delta d_{i,j}^{\Delta t} \) can be calculated as

\[
\Delta d_{i,j}^{\Delta t} = \begin{cases} 
  d_x - d_{i,j}^{t_1}, & \text{Case 1,} \\
  d_x + d_{i,j}^{t_1}, & \text{Case 2,}
\end{cases}
\] (3.10)

where Case 1 and Case 2 are the same as defined in above. As \( d_{i,j}^{t_1} > d_x \), for \( d_{i,j}^{t_2} \leq d_x \), it is required that \( \Delta d_{i,j}^{\Delta t} < 0 \). Therefore, \( 1 - f(\Delta d_{i,j}^{\Delta t}) \) is calculated and ultimately

\[
Pr(d_{i,j}^{t_2} \leq d_x) = Pr(SINR_{i,j}^{t_2} \geq \beta_1) \] is expressed as

\[
Pr(SINR_{i,j}^{t_2} \geq \beta_1) = 1 - \frac{1}{2} \left( \text{erf} \left( \frac{\Delta d_{i,j}^{\Delta t}}{2(\sigma_i^2 + \sigma_j^2)\Delta t^3} \right) - \text{erf} \left( \frac{-\Delta d_{i,j}^{\Delta t}}{2(\sigma_i^2 + \sigma_j^2)\Delta t^3} \right) \right).
\] (3.11)
\(Q(SINR_{i}^{t_2}) = \sum_{j=1}^{n_{\text{neigh}}} \Pr(SINR_{i,j}^{t_2} \geq \beta_1)\), where \(n_{\text{neigh}}\) is the number of neighbours of node \(i\) whose position and velocities are exchanged through beacon messages.

\(b)\ \ DF_{i}^{t_2}\)

It is the probability that one hop distance between node \(i\) and the sender \(s\) is larger than a minimum threshold, \(d_{h_{\text{op}}}^{\text{min}}\), and is defined as

\[DF_{i}^{t_2} = \Pr(d_{i,s}^{t_2} > d_{h_{\text{op}}}^{\text{min}}) = 1 - \Pr(d_{i,s}^{t_2} \leq d_{h_{\text{op}}}^{\text{min}}), \quad (3.12)\]

where \(\Pr(d_{i,s}^{t_2} \leq d_{h_{\text{op}}}^{\text{min}})\) can be found by using the same calculation as described in proof of Theorem 3.1.

- Proposition 3.1: \(0 \leq QF_{i} \leq n_{\text{neigh}}\).

Proof: \(Q(SINR_{i}^{t_2})\) is a sum of \(\Pr(SINR_{i,j}^{t_2})\), for all neighbor nodes of \(i\). Therefore, the possible range of \(Q(SINR_{i}^{t_2})\) is \(0 \leq Q(SINR_{i}^{t_2}) \leq n_{\text{neigh}}\). According to (3.12), the possible range of \(DF_{i}^{t_2}\) is \(0 \leq DF_{i}^{t_2} \leq 1\). As \(QF_{i} = Q(SINR_{i}^{t_2}) \cdot DF_{i}^{t_2}\), it can be concluded that \(0 \leq QF_{i} \leq n_{\text{neigh}}\).  

3.2.4 Adversary Model

Let \(n_{mn}\) be the number of mining nodes in a PoQF consensus out of which \(n_{m}\) nodes are malicious and \(n_{h}\) nodes are honest when \(n_{h} = 1\). A malicious node in the proposed blockchain design is defined as the node voting against the original validity of a message, that is, if a message is true, the malicious node will vote false and vice-versa. Let \(B_{i}\) be the behavior of mining node \(i\). \(B_{i} = 1\) when it is malicious and \(B_{i} = 0\) when it is honest and \(n_{m} = \sum_{i=1}^{n_{mn}} B_{i}\). \(B_{i}\) follows Binomial distribution, i.e., \(B_{i} \sim \mathcal{B}(n_{mn}, p_{m})\), where \(p_{m}[0,1]\) is the probability that \(B_{i} = 1\). The reason for considering Binomial distribution is because it has only two possible outcomes for a discrete random number. So, one of the outcomes can be defined as malicious and another as honest. \(\mu_{m} = p_{m}n_{mn}\) and \(\mu_{h} = (1 - p_{m})n_{mn}\) represent the mean number of malicious and honest nodes, respectively.
3.2.5 Incentive Distribution Mechanism

Call Compensation ($CC = CC_{mn} + CC_r$) is the virtual credit paid by ORG as a penalty to cause incident and compensate affected nodes. As shown in Figure 3.4, assuming that a message is successfully validated, $CC$ at each direction consists of $CC_{mn}$, which is equally distributed among honest mining nodes at $n_{hop} = 1$, and $CC_r$, which is equally distributed among RLYs at $n_{hop} = \{1,2,\ldots,n_{hop}^{\max}\}$, in case a message is validated as true. Otherwise, $CC_r$ is transferred to CA as a penalty charge. The values of $CC_{mn}, CC_r$ and $n_{hop}^{\max}$ are specified by CA and communicated to nodes via RSU.

![Figure 3.4: Distribution of Call Compensation.](image)

a) **Reputation**

Let $Rep_i$ be the reputation of node $i$. A node $i$ is eligible to become a mining node and broadcast its vote if $Rep_i > Rep_T$, where $Rep_T$ is a certain threshold. It is proposed that RLY is responsible to store updated $Rep_i$ of each node $i$ into the blockchain, whenever it performs block addition. Also, $CC$ paid by ORG is inversely proportional to its reputation, i.e., $CC \propto \frac{1}{Rep_{ORG}}$.

Let $Rep_{Rew}$ be the amount of reputation rewarded for honest behaviour and deducted for malicious behaviour. The updated reputation of mining node $i$ after PoQF consensus at $n_{hop} = 1$ is
\[Rep_l = \begin{cases} \frac{CC_{mn}}{n_h}, & B_l = 0 \text{ and message is true,} \\ \frac{CC_{mn}}{n_h} - TC, & B_l = 0 \text{ and message is false,} \\ -TC, & B_l = 1 \text{ and message is true,} \\ 0, & B_l = 1 \text{ and message is false,} \end{cases}\] (3.14)

where \(TC > 0\) is the Transaction Charge paid by mining node \(i\) only when it votes that a message is false. It is later paid to the \(RLY\) which generates the last keyblock related to a particular incident. The motivation of introducing \(TC\) is only to vote false is to promote fast dissemination of true message in case of emergency. The utility of \(RLY\), \(U_{RLY}\), is given as

\[U_{RLY} = \begin{cases} \frac{CC_{r}}{n_{hop}}, & n_{hop} \leq n_{hop}^{max} \text{ and message is true}, \\ n_{m}TC, & n_{hop} > n_{hop}^{max} \text{ and message is true}, \\ n_{h}TC, & n_{hop} = 1 \text{ and message is false}, \\ 0, & \text{otherwise.} \end{cases}\] (3.15)

It is worth noting that a mining node \(i\) at \(n_{hop} = 1\) selected as \(RLY\) will earn a cumulative utility of \(U_i + U_{RLY}\). \(RLY\) records transactions related to \(U_i\) in the keyblock at \(n_{hop} = 1\). For \(n_{hop} > 1\), the corresponding \(RLY\) records transaction related to \(U_{RLY}\) of previous hop.
The message is disseminated until \( n_{hop} \leq n_{hop}^{max} \) and PoQF is repeated until \( n_{hop} \leq n_{hop}^{max} + 1 \), because the last RLY at \( n_{hop} \leq n_{hop}^{max} + 1 \) records \( U_{RLY} \) of RLY at \( n_{hop}^{max} \). As an incentive, it gains the reward of \( n_m TC \) and records this transaction itself.

\[ c) \text{ Mechanism 2: TC for all transmissions} \]

Let \( TC \) be paid by every node which broadcasts a message or vote. The purpose of introducing \( TC \) for every transmission is to demotivate nodes to cast a fake vote or disseminate false message at the expense of their virtual credit. If the message is successfully validated, the utility of a mining node \( i \), \( U_i \), after taking part in a PoQF consensus at \( n_{hop} = 1 \) is given as

\[
U_i = \begin{cases} 
\frac{CC_{mn}}{n_h} - TC, & B_i = 0, \\
-TC, & B_i = 1,
\end{cases}
\]

(3.16)

where \( CC_{mn} \) is the portion of \( CC \) distributed among mining nodes. Considering message dissemination in only one direction, let \( CC = CC_{mn} + CC_r \), where \( CC_{mn} = \omega_1 CC \), \( CC_r = \omega_2 CC \), \( \omega_1 \) and \( \omega_2 \) are weights to divide \( CC \) among mining nodes and RLYs respectively and \( \omega_1 + \omega_2 = 1 \). The utility of RLY is,

\[
U_{RLY} = \begin{cases} 
\frac{CC_r}{n_{hop}^{max}} - TC, & \text{message is true}, \\
0, & \text{otherwise},
\end{cases}
\]

(3.17)

\( TC \) paid by every node is deposited to \( CA \). \( CA \) regulates \( TC \) according to the incident recovery cost estimated at the location of incident and time of the day.

3.3 Theoretical Analysis

3.3.1 Security

\[ a) \text{ Failure in Validation (FV)} \]

In this chapter, the term \( FV \) is defined as the failure in determining the correct validity status of a message by PoQF consensus at \( n_{hop} = 1 \). Without loss of generality, it is assumed the probability that an \( ORG \) generates a false message, i.e., the \( ORG \) is malicious,
is $p_m$ and the probability of true message generation is $1 - p_m$. Therefore, $FV$ can be expressed as

$$FV = p_m FV_{false} + (1 - p_m) FV_{true}, \quad (3.18)$$

where $FV_{false}$ and $FV_{true}$ denote $FV$ of false and true message respectively. $FV_{false}$ occurs when a malicious mining node receives at least $n_{th}$ microblocks with malicious votes to mark an originally false message as true. Therefore, $FV_{false}$ can be given as

$$FV_{false} = p_m P_r(n_m \geq n_{th}). \quad (3.19)$$

$FV_{true}$ occurs when an honest mining node does not receive $n_{th}$ microblocks with honest votes to validate an originally true message and can be expressed as

$$FV_{true} = 1 - (1 - p_m) P_r(n_h \geq n_{th}). \quad (3.20)$$

Bringing (3.19) and (3.20) into (3.18) gives

$$FV = 1 - p_m + p_m^2 P_r(n_m \geq n_{th}) - (1 - p_m)^2 P_r(n_h \geq n_{th}). \quad (3.21)$$

The following propositions are proved using tail inequalities for Binomial distribution [121].

- **Proposition 3.2**: The upper bound of $P_r(n_x \geq n_{th})$, where $x = m$ or $h$ can be given as

$$P_r(n_x \geq n_{th})^{UB} = \begin{cases} e^{-\frac{(n_{th} - \mu_x)^2}{n_{th} + \mu_x}}, & n_{th} \geq \mu_x \\ 1, & \text{otherwise}. \end{cases}$$

**Proof**: According to the multiplicative form of Chernoff bound [121], $P_r(X \geq (1 + \delta)\mu) \leq e^{-\frac{\delta^2 \mu}{2 + \delta}}$, where $X$ is a sum of independent Binomial variables with mean $\mu$ and $\delta > 0$. Bringing $\mu = \mu_x$ and $(1 + \delta)\mu = n_{th}$ in Chernoff bound inequality gives
• **Proposition 3.3:** The lower bound of $P(n_x \geq n_{th})$, where $x = m$ or $h$ can be given as

$$P(n_x \geq n_{th})^{LB} = \begin{cases} 
1 - e^{-\frac{(\mu_x - n_{th})^2}{2\mu_x}}, & n_{th} \geq \mu_x \\
0, & \text{otherwise.}
\end{cases}$$

Proof: For $0 \leq \delta \leq 1$, Chernoff bound [121] states that $P(X \leq (1 - \delta)\mu) \leq e^{-\frac{\delta^2 \mu}{2}}$, which can be rewritten as $P(X \geq (1 - \delta)\mu) \geq 1 - e^{-\frac{\delta^2 \mu}{2}}$. Therefore, bringing $\mu = \mu_x$ and $(1 - \delta)\mu = n_{th}$ in Chernoff bound inequality gives

$$P(n_x \geq n_{th}) \geq \begin{cases} 
1 - e^{-\frac{(\mu_x - n_{th})^2}{2\mu_x}}, & n_{th} \geq \mu_x \\
1, & \text{otherwise.}
\end{cases}$$

By applying Proposition 3.2 and Proposition 3.3 in (3.21), the upper and lower bounds of $FV$, $FV^{UB}$ and $FV^{LB}$ can be derived as

$$FV^{UB} = 1 - p_m + p_m^2 P(n_m \geq n_{th})^{UB} - (1 - p_m)^2 P(n_h \geq n_{th})^{LB}, \quad (3.24)$$

$$FV^{LB} = 1 - p_m + p_m^2 P(n_m \geq n_{th})^{LB} - (1 - p_m)^2 P(n_h \geq n_{th})^{UB}. \quad (3.25)$$

The expanded forms of (3.24) and (3.25) under a varying range of $n_{th}$ can be derived using (3.21), (3.22) and (3.23), as follows

$$FV \leq \begin{cases} 
1 - p_m + p_m^2 - (1 - p_m)^2 (1 - e^{-\frac{(\mu_h - n_{th})^2}{2\mu_h}}), & n_{th} < \min(\mu_m, \mu_h), \\
0, & \mu_m \leq n_{th} \leq \mu_h, \\
1 - p_m + p_m^2 e^{-\frac{(n_{th} - \mu_m)^2}{\mu_m + n_{th}}} - (1 - p_m)^2 (1 - e^{-\frac{(\mu_h - n_{th})^2}{2\mu_h}}), & \mu_h < n_{th} < \mu_m, \\
1 - p_m + p_m^2 e^{-\frac{(n_{th} - \mu_m)^2}{\mu_m + n_{th}}}, & n_{th} > \max(\mu_m, \mu_h),
\end{cases}$$

(3.26)
and,

\[
FV \geq \begin{cases} 
1 - p_m + p_m^2 (1 - e^{-\frac{(\mu_m - n_{th})^2}{2\mu_m}}) - (1 - p_m)^2, & n_{th} < \min(\mu_m, \mu_h), \\
p_m - p_m^2, & \mu_m \leq n_{th} \leq \mu_h, \\
1 - p_m + p_m^2 (1 - e^{-\frac{(\mu_m - n_{th})^2}{2\mu_m}}) - (1 - p_m)^2 e^{-\frac{(n_{th} - \mu_h)^2}{\mu_h + n_{th}}}, & \mu_h < n_{th} < \mu_m, \\
1 - p_m - (1 - p_m)^2 e^{-\frac{(n_{th} - \mu_h)^2}{\mu_h + n_{th}}}, & n_{th} > \max(\mu_m, \mu_h). 
\end{cases}
\]

(3.27)

• **Proposition 3.4:** Minimum $FV^{UB}$ at $p_m > 0.5$ can be obtained when $n_{th} > \mu_m$.

**Proof:** To find the minimum $FV^{UB}$, we compare its value at two conditions of (3.26), i.e., $n_{th} > \max(\mu_m, \mu_h)$ and $n_{th} < \min(\mu_m, \mu_h)$, which are apparently less than others.

\[
1 - p_m + p_m^2 e^{-\frac{(n_{th} - \mu_m)^2}{\mu_m + n_{th}}} < 1 - p_m + p_m^2 - (1 - p_m)^2 \left(1 - e^{-\frac{(\mu_h - n_{th} - 1)^2}{2\mu_h}}\right),
\]

which can be simplified as

\[
p_m^2 e^{-\frac{(n_{th} - \mu_m)^2}{\mu_m + n_{th}}} < 2p_m - 1 - (1 - p_m)^2 \left(1 - e^{-\frac{(\mu_h - n_{th} - 1)^2}{2\mu_h}}\right).
\]

(3.29)

Assuming that $p_m^2 e^{-\frac{(n_{th} - \mu_m)^2}{\mu_m + n_{th}}} \approx (1 - p_m)^2 \left(1 - e^{-\frac{(\mu_h - n_{th} - 1)^2}{2\mu_h}}\right) \approx 0$, (3.29) leads towards the condition, $p_m > 0.5$. It shows that the condition in (3.32) is fulfilled when $p_m > 0.5$ and therefore, according to (3.26), minimum $FV^{UB}$ is obtained when $n_{th} > \max(\mu_m, \mu_h)$. Since $\mu_m > \mu_h$ only when $p_m > 0.5$, it means that minimum $FV^{UB}$ at $p_m > 0.5$ is attained when $n_{th} > \mu_m$.

• **Proposition 3.5:** Minimum $FV^{LB}$ can be obtained when $\mu_m \leq n_{th} \leq \mu_h$ at $p_m \leq 0.5$ and when $n_{th} > \mu_m$ at $p_m > 0.5$. 

**Proof:** As we know that, \(0 < e^{-x} \leq 1\) for any real valued \(x\) and \(p_m \epsilon [0,1]\), it can be deduced from (3.27) that \(FV^{LB}\) is the minimum when \(\mu_m \leq n_{th} \leq \mu_h\), which is only possible for \(p_m \leq 0.5\). For \(p_m > 0.5\), the minimum \(FV^{LB}\) can be obtained when \(n_{th} > \mu_m\).

\[\Box\]

**Forks**

\(b)\) **Forks**

![Flowchart of actions to resolve forks](image)

**Figure 3.5:** Resolving forks in the proposed PoQF consensus.

In the proposed blockchain, a fork may be created as shown in Figure 3.5 (a) when two keyblocks are generated by different \(RLYs\) at the same hop. Forks occur when two or more nodes fulfil both conditions of becoming a \(RLY\), which are defined in Section 3.2. The flowchart of actions by a node in case of fork occurrence is shown in Figure 3.5 (b). If the keyblock by \(RLY\) \(i\) marks the message as false and the keyblock by \(RLY\) \(j\) marks the message as true, then the message dissemination is continued and new blocks are linked.
with the keyblock generated by \( RLY \, j \). If both nodes show the same validity and \( QF_i = QF_j \), the timestamps of both keyblocks are checked and the keyblock with the earlier timestamp is considered valid. However, if \( QF_i > QF_j \), then new blocks are added in continuation with the keyblock generated by \( RLY \, i \), regardless of the timestamp of \( RLY \, j \). The motivation behind selecting the keyblock on the basis of \( QF_i \) instead of timestamp for blockchain extension is to discourage a possible cheating attempt by mining node \( j \) to become a \( RLY \) despite having \( QF_i < QF_j \). Cheating by manipulating \( QF_i \) is difficult, as it is based on position and velocity of nodes which are shared through regular beacon message exchange and a cheating attempt can be easily detected and reported to concerned authority. In presence of forks, only the longest chain is stored.

c) Game Theory Analysis of Incentive Distribution Mechanism 1: TC for only voting towards false messages

We apply the game theory to analyse the impact of the proposed incentive distribution mechanism on actions of mining nodes at \( n_{hop} = 1 \) and evaluate the security of PoQF against colluding attack by mining nodes.

- **Players**: This game has \( n_{mn} \) players out of which \( n_h \) are honest and \( n_m \) are malicious. All players follow PoQF consensus as mining nodes and are located at \( n_{hop} = 1 \).
- **Actions**: Every player has two possible actions, honest, \( H \), or malicious, \( M \).
- **Utilities**: The payoff matrix in Table 3.2 shows \((U_i, U_y)\), if \( FV \) does not occur after PoQF at \( n_{hop} = 1 \).

The analysis of the proposed incentive distribution mechanism is as follows.

- **Lemma 3.2**: Playing honest is the best response action of a mining node, if \( CC_{mn} \geq n_h TC \).

**Proof**: As shown in Table 3.2, if \( TC \geq \frac{CC_{mn}}{n_h} \) and the message is false, playing honest will result in \( U_y \leq 0 \) which will be motivated to play maliciously. On the contrary, if \( TC \leq \frac{CC_{mn}}{n_h} \), it makes \( U_y \geq 0 \) which will motivate the mining nodes to play honestly.
Therefore, to make honest as the best response action of mining nodes, it is required that \( TC \leq \frac{CC_{mn}}{n_h} \) or \( CC_{mn} \geq n_h TC \).  

<table>
<thead>
<tr>
<th>Mining node i with the highest ( QF_i )</th>
<th>Any other mining node</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( H )</td>
<td>( M )</td>
<td></td>
</tr>
<tr>
<td>( H )</td>
<td>( \left( \frac{CC_{mn}}{n_h} + \frac{CC_r}{n_{hop}^{max}}, \frac{CC_{mn}}{n_h} \right) )</td>
<td>( \left( \frac{CC_{mn}}{n_h} + \frac{CC_r}{n_{hop}^{max}}, -TC \right) )</td>
<td></td>
</tr>
<tr>
<td>( M )</td>
<td>( (-TC, \frac{CC_{mn}}{n_h}) )</td>
<td>( (-TC, -TC) )</td>
<td></td>
</tr>
</tbody>
</table>

(a) True Message

<table>
<thead>
<tr>
<th>Mining node i with the highest ( QF_i )</th>
<th>Any other mining node</th>
<th>( H )</th>
<th>( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H )</td>
<td>( \left( \frac{CC_{mn}}{n_h} + (n_h - 1)TC, \frac{CC_{mn}}{n_h} - TC \right) )</td>
<td>( (n_h - 1)TC, 0 )</td>
<td></td>
</tr>
<tr>
<td>( M )</td>
<td>( \left( 0, \frac{CC_{mn}}{n_h} - TC \right) )</td>
<td>( (0,0) )</td>
<td></td>
</tr>
</tbody>
</table>

(b) False Message

Table 3.2: Payoff Matrix \((U_i, U_y)\), where \( U_i = \) Utility of mining node \( i \) with the highest \( QF_i \) and \( U_y = \) Utility of any other mining node at \( n_{hop} = 1 \).

- **Proposition 3.6**: The action set \((H, H)\) is both Pareto-optimal and Nash equilibrium of this game.

**Proof**: From the payoff matrix in Table 3.2, it can be seen that no player can get the maximum utility by deviating from the action set \((H, H)\), provided that Lemma 3.2 is fulfilled. In both true and false message cases, all mining nodes can get the highest payoff by playing honestly only. Therefore, the action set \((H, H)\) is both Pareto-optimal and Nash equilibrium of this game.  

\[ \square \]
Theorem 3.2: A mining node cannot increase its expected utility sum by colluding with its malicious neighbours if \( n_h T_C \leq CC_{mn} \leq \frac{n_m CC_r}{n_{hop}(n_h-n_m)} \) and \( p_m \leq 0.5 \).

Proof: Let \( n_{cp} \) colluding players form a group to mark a true message as false or a false message as true with a probability \( p_{cp} \). The expected utility sum of colluding players as mining nodes, \( E(U_{cp}) \), if they mark a true message as false is

\[
E(U_{cp}) = p_{cp} \left( \frac{CC_{mn}}{n_m} n_{cp} - n_{cp} T_C \right) + \left(1 - p_{cp}\right) \left( \frac{CC_{mn}}{n_h} - n_{cp} \right).
\]

(3.30)

The probability that one of the colluding players is selected as a RLY if the colluding attack is successful is \( n_{cp}/n_m \) and if colluding players play honestly is \( n_{cp}/n_h \). Therefore, the total expected utility sum \( E(U_{cp}) \) is given as

\[
E(U_{cp}) = p_{cp} \left( \frac{CC_{mn}}{n_m} n_{cp} - n_{cp} T_C + \frac{n_{cp}}{n_m} n_{cp} T_C \right)
\]

\[
+ \left(1 - p_{cp}\right) \left( \frac{CC_{mn}}{n_h} n_{cp} + \frac{n_{cp}}{n_h} \frac{CC_r}{n_{hop}} \right).
\]

(3.31)

To prevent collusion, it is required that \( E(U_{cp}) \leq E(U'_{cp}) \), where \( U'_{cp} \) represents the utility of colluding players playing honestly, i.e.,

\[
p_{cp} \left( \frac{CC_{mn}}{n_m} n_{cp} \right) + \left(1 - p_{cp}\right) \left( \frac{CC_{mn}}{n_h} n_{cp} + \frac{n_{cp}}{n_h} \frac{CC_r}{n_{hop}} \right)
\]

\[
\leq \frac{CC_{mn}}{n_h} n_{cp} + \frac{n_{cp}}{n_h} \frac{CC_r}{n_{hop}},
\]

(3.32)

which leads towards the condition, \( CC_{mn} \leq \frac{n_m CC_r}{n_{hop}(n_h-n_m)} \). If \( CC_{mn} \geq 0 \), this condition can only be fulfilled when \( n_h \geq n_m \), i.e., when \( p_m \leq 0.5 \). Similarly, if colluding players attempt to mark a false message as true, then \( E(U_{cp}) \) is given as,
\[
E(U_{cp}) = p_{cp} \left( \frac{CC_{mn}}{n_m} n_{cp} + \left( \frac{n_{cp}}{n_m} \right) \frac{CC_r}{n_{hop}^{\max}} \right) \\
+ \left( 1 - p_{cp} \right) \left( \frac{CC_{mn}}{n_h} n_{cp} - n_{cp} TC + \left( \frac{n_{cp}}{n_h} \right) n_h TC \right).
\] (3.33)

To prevent collusion, it is required that \( E(U_{cp}) \leq E(U'_{cp}) \), which leads towards the condition \( CC_{mn} \geq \frac{n_h CC_r}{n_{hop}^{\max} (n_m - n_h)} \). Combining the condition of Lemma 3.2, it is needed that, \( CC_{mn} \geq \max \left( n_h TC, \frac{n_h CC_r}{n_{hop}^{\max} (n_m - n_h)} \right) \). For \( p_m \leq 0.5 \), \( CC_r > 0 \) and \( n_{hop}^{\max} > 0 \), we always get \( \frac{n_h CC_r}{n_{hop}^{\max} (n_m - n_h)} \leq 0 \) and \( n_h TC \geq \frac{n_h CC_r}{n_{hop}^{\max} (n_m - n_h)} \). Therefore, it is proved that the incentive distribution mechanism is collusion resistant if \( n_h TC \leq CC_{mn} \leq \frac{n_m CC_r}{n_{hop}^{\max} (n_h - n_m)} \) and \( p_m \leq 0.5 \).

Thus, the incentive distribution mechanism is collusion resistant if values of \( CC_{mn} \), \( CC_r \) and \( n_{hop}^{\max} \) are adjusted such that Theorem 3.2 is fulfilled.

d) **Game Theory Analysis of Incentive Distribution Mechanism 2: TC for all transmissions**

The following proposition and theorem are presented to prove security of the incentive distribution mechanism against collusion.

- **Lemma 3.3**: Playing honest is the best response action of all players if \( TC < \min \left( \frac{CC_{mn}}{n_h}, \frac{CC_r}{n_{hop}^{\max}} \right) \).

**Proof**: According to (3.16), if \( TC < 0 \), \( U_i \) would be positive if a mining node \( i \) plays maliciously. On the other hand, if \( TC \geq \frac{CC_{mn}}{n_h} \), \( U_i \) will be zero or negative if it plays honestly. In either of these cases, the best response action of players would be to play maliciously or selfishly. Also, according to (3.17), if \( TC \geq \frac{CC_r}{n_{hop}^{\max}} \), \( U_{RLY} \) will be negative and \( RLY \) will be motivated to play selfishly. Therefore \( TC \) must be set such that the
action results in positive $U_i$ and $U_{RLY}$ or profit gain of both mining node $i$ and $RLY$.

- **Theorem 3.3:** The proposed solution motivates all players to play honestly and is resistant against collusion of $RLY$s if $0 \leq TC < \min\left(\frac{CC_{mn}}{n_h}, \frac{CC_r}{n_{hop}^{\max}}\right)$.

**Proof:** Let us consider the case with one conspired $RLY$ at $n_{hop}$, i.e., $RLY(n_{hop})$. Suppose $CP = \{RLY(n_{hop}), RLY(n_{hop}+1)\}$ is a group of colluding players. $CP$ conspires to form a longer path with an additional hop, i.e., $ORG \rightarrow RLY(n_{hop}) \rightarrow RLY(n_{hop}+1)$ instead of the most appropriate path, i.e., $ORG \rightarrow RLY'(n_{hop}+1)$. In both cases, the message reaches a maximum specified distance. Let $p_{cp}$ be the probability with which $RLY(n_{hop})$ encounters $RLY(n_{hop}+1)$, where $p_{cp} \in [0,1]$ and $p_{cp}^2$ to encounter both $ORG$ and $RLY(n_{hop}+1)$. The expected utility sum of $CG$, $E(U_{cp})$ is

$$E(U_{cp}) = p_{cp}^2 \left(U_{RLY(n_{hop})} + U_{RLY(n_{hop}+1)}\right) + (1 - p_{cp}^2)U_{RLY'(n_{hop}+1)}, \tag{3.34}$$

or,

$$E(U_{cp}) = p_{cp}^2 \left(\frac{2CC_r}{n_{hop}^{\max}} - 2TC\right) + (1 - p_{cp}^2) \left(\frac{CC_r}{n_{hop}^{\max}} - TC\right), \tag{3.35}$$

Since $n_{hop}^{\max} = 2$ for collusion case and $n_{hop}^{\max} = 1$ for non-collusion, $E(U_{cp})$ becomes

$$E(U_{cp}) = CC_r - TC - p_{cp}^2TC. \tag{3.36}$$

To avoid collusion of relay nodes, it is required that $E(U_{cp}) \leq U_{RLY'(n_{hop}+1)}$, that is,

$$CC_r - TC - p_{cp}^2TC \leq CC_r - TC. \tag{3.37}$$

It follows that

$$-p_{cp}^2TC \leq 0, \tag{3.38}$$
or $TC \geq 0$. Similarly, by generalizing cases when $n_{cp} > 2$, the collusion resistant condition can be derived, that is, $p_{cp}^{n_{cp}}TC \geq 0$. Hence, for any $p_{cp} \in [0,1]$, the proposed incentive distribution mechanism is relay node collusion resistant if $TC \geq 0$. The upper bound of $TC$ required to motivate all players to play honestly, i.e., $TC < \min\left(\frac{CC_{mn}}{n_h}, \frac{CC_{r}}{n_{hop}}\right)$ is proved in Lemma 3.3. Therefore, the proposed incentive mechanism is secure against malicious actions and collusion if $0 \leq TC < \min\left(\frac{CC_{mn}}{n_h}, \frac{CC_{r}}{n_{hop}}\right)$. ■

3.3.2 Latency and Throughput

The MAC throughput in bit/second is defined in [122] as $\lambda_{MAC} = p_t \cdot p_{suc} \cdot \frac{L}{T_{avg}}$, where $p_t = \frac{2}{W+2}$ is the average transmission probability of a node, $W$ is the contention window size, $p_{suc}$ is the probability of success transmission, $L$ is the average length of a packet and $T_{avg}$ is the average length of a time slot in Distributed Coordination Function (DCF). $p_{suc} = n_{tr} \cdot p_t \cdot (1-p_t)^{n_{tr}-1}$, where $n_{tr}$ is the number of nodes contending the channel for transmission. According to IEEE 802.11 standard [122], $T_{avg}$ is

$$T_{avg} = p_{idle} \cdot T_{slot} + p_{suc} \cdot T_{suc} + p_{col} \cdot T_{col}, \quad (3.39)$$

where $T_{slot}$ is the unit time slot in DCF scheme, $p_{idle} = (1-p_t)^{n_{tr}}$ and $p_{col} = 1 - p_{idle} - p_{suc}$ are the probabilities of a node encountering an idle slot and collided transmission respectively, $T_{suc}$ and $T_{col}$ are average time for success and collided transmission respectively and are given as $T_{col} = T_{RTS} + T_{DIFS} + T_{slot}$ and $T_{suc} = T_{RTS} + T_{DIFS} + T_{CTS} + T_{ACK} + 3T_{SIFS} + 4T_{slot} + \frac{L}{DR}$, where $T_{DIFS}$ and $T_{SIFS}$ are time intervals for DCF Interframe space (DIFS) and Short Interframe Space (SIFS) respectively, $T_{RTS}$, $T_{CTS}$ and $T_{ACK}$ are pre-specified time intervals reserved for DCF related operations and $DR$ is the average data rate among nodes. As $\lambda_{MAC}$ is defined in bit/second, the average time consumption, $T_{MB}$, to successfully transmit a vote in a microblock of length $L$ bits is

$$T_{MB} = \frac{L\text{ (bits)}}{\lambda_{MAC}\text{ (bit/second)}} = \frac{T_{avg}}{p_t p_{suc}}, \quad (3.40)$$
$T_{MB}$ varies with $n_{tr}$ only if $W$, $L$, $T_{slot}$, $T_{col}$ and $T_{suc}$ are considered as fixed for all transmitting mining nodes. As $1 \leq n_{tr} \leq n_{mn}$ and $n_{mn} = n_{neigh}$ if all neighbour nodes of ORG take part in consensus. If $n_{neigh}$ is estimated by the number of received beacons [118] or with the use of edge computing servers [119], a node $i$ can find $T_{MB}^{min}$ and $T_{MB}^{max}$. Considering a fixed transmission range and a homogeneous distribution for all nodes, it can be assumed that $n_{neigh}$ is statistically the same for every node. According to Proposition 3.1, $n_{neigh} - QF_i$ can be considered as the ranking of mining node $i$ to announce its microblock. In this way, node $i$ with a large $QF_i$ can have less validation time before announcing a microblock. Consider $T_{MB}$ as the time required by a mining node to successfully transmit a microblock. The lower bound of $\tau_i$ is given as

$$a_{\tau_i} = T_{MB}^{min}(n_{neigh} - QF_i),$$

and the upper bound of $\tau_i$ is given as

$$b_{\tau_i} = T_{MB}^{max}(n_{neigh} - QF_i),$$

For $n_{mn}$ microblocks, the total time consumption (or validation latency), $T_{delay}$ can be in the range $T_{MB}^{min} \cdot n_{mn} \leq T_{delay} \leq T_{MB}^{max} \cdot n_{mn}$. A mining node $i$ with the highest $QF_i$ becomes $RLY$ as soon as it receives at least $n_{th}$ microblocks and does not need to wait for receiving all $n_{mn}$ microblocks. Therefore, $T_{delay}$ is reduced for small $n_{th}$ and lower and upper bounds of $T_{delay}$ when a message is successfully validated by an honest node at $n_{hop} = 1$ can be found.

- **Proposition 3.7:** $T_{delay}^{LB} = T_{MB}^{min} \cdot n_{th}$.

  **Proof:** $T_{delay}$ is the minimum when $RLY$ receives first $n_{th}$ consecutive microblocks with same votes immediately following the incident message. ■

- **Proposition 3.8:** $T_{delay}^{UB} = T_{MB}^{max} \cdot (n_{th} + p_m n_{mn})$.

  **Proof:** The maximum number of microblocks with malicious votes is $p_m n_{mn}$. $T_{delay}$ will be the maximum if an honest $RLY$ receives all $p_m n_{mn}$ microblocks before receiving $n_{th}$ honest microblocks. ■
A keyblock is generated by $RLY$ after the message validation. Therefore, the throughput in terms of number of keyblocks generated per second can be estimated as

$$\lambda_{KB} = \frac{1}{T_{delay} + T_{eyp}},$$  \hspace{1cm} (3.43)

where $T_{eyp}$ is the time required to encrypt a keyblock.

### 3.3.3 Asymptotic Complexities

<table>
<thead>
<tr>
<th>Consensus</th>
<th>Latency</th>
<th>Security</th>
<th>Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoW</td>
<td>$\Theta(\kappa)$ [123]</td>
<td>$\Omega\left(\frac{n_{mn}}{2}\right)$ [124]</td>
<td>$\Theta(1)$ [123]</td>
</tr>
<tr>
<td>PoS</td>
<td>$\Omega(\kappa)$ [123]</td>
<td>$\Omega\left(\frac{2n_{mn}}{3}\right)$ [125]</td>
<td>$\Theta(1)$ [123]</td>
</tr>
<tr>
<td>PoET</td>
<td>$\Omega(\kappa)$ [78]</td>
<td>$\Omega\left(\frac{\log \log n_{mn}}{\log n_{mn}}\right)$ [78]</td>
<td>$\Theta(1)$ [78]</td>
</tr>
<tr>
<td>PBFT</td>
<td>$n_{mn}O(1)$ [123]</td>
<td>$\Omega\left(\frac{n_{mn} - 1}{2}\right)$ [113]</td>
<td>$O(n_{mn}^2)$ [126]</td>
</tr>
<tr>
<td>PoQF</td>
<td>$\kappa O(1)$</td>
<td>$\Omega\left(\frac{n_{mn}}{2}\right)$</td>
<td>$O(n_{mn})$</td>
</tr>
</tbody>
</table>

Table 3.3: Comparison of asymptotic complexities.

In this subsection, the scalability of various consensus algorithms are compared by analysing latency complexity, i.e., the time consumption required to confirm a transaction, security complexity, i.e., the minimum number of malicious nodes to control consensus, and communication complexity, i.e., the number of exchange messages required to validate a transaction. Without loss of generality, the asymptotic latency, security and communication complexity of various consensus algorithms are identified in Table 3.3 in terms of number of nodes participating in mining competition, $n_{mn}$ and consensus parameter, $\kappa$, which is unique to each algorithm. $\kappa$ refers to the difficulty level of hash puzzle in PoW, synchronization level in PoS, waiting time in PoET and number of minimum votes required in voting based algorithms (PBFT and PoQF). Standard mathematical notations are used in Table 3.3, i.e., $\Omega(.)$, $O(.)$ and $\Theta(.)$ denote the order of at least, at most and exactly respectively.
Table 3.3 shows that $\kappa$ affects the latency in PoW, PoS and PoET. Despite the fast consensus of PoS, a strong synchronization among edge computing resources is needed for efficient running [123]. Latency of PoET depends on the length of waiting time which follows a fixed probability distribution. PoQF has to wait for a threshold number of votes, which has an impact on latency but its scalability does not rely on large computation power or storage capacity of nodes. PoET offers the least security and can be controlled by only a small fraction of malicious nodes [78]. According to Theorem 3.2, PoQF is secure against the collusion attack when $p_m \leq 0.5$. It provides the same security as PoW which is better than PBFT but worse than PoS [125]. In communication complexity, PoW, PoS and PoET are more efficient than PoQF, since they do not require multiple message exchanges. Despite the voting nature of PoQF, it has lower communication complexity than PBFT. Moreover, in VANETs, $n_{mn}$ cannot be increased beyond a certain threshold due to limited number of nodes within a transmission range $R$ and $d_{min}^{neigh}$, which makes PoQF scalable and applicable in V2V communications.

3.4 Simulated Performance Analysis

3.4.1 Simulation Setup

In this section, the performance of the proposed blockchain and PoQF consensus is analysed using OMNeT++ and SUMO (Simulation of Urban Mobility). An open-source framework VeINS (Vehicles In Network Simulation) is used to integrate SUMO with OMNeT++\textsuperscript{1}. The code of blockchain implementation in OMNeT++ is mentioned in Appendix B. The simulation parameters listed in Table 3.4. Since $n_{mn}$ are neighbor nodes of a sender $s$, $n_{mn} \leq 40$ will be considered when nodes are homogeneously distributed with a maximum of 200 nodes/km$^2$ and it is a reasonable assumption of maximum number of vehicles within a transmission range when the safe distance between nodes are maintained on road. Evaluation results are averaged over 100 simulation runs.

\textsuperscript{1} Source code is available at https://zenodo.org/record/5172646#.YW16F_rMKU1.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>200 s</td>
<td>Protocol</td>
<td>IEEE 802.11p</td>
</tr>
<tr>
<td>Size of area</td>
<td>10 km × 10 km</td>
<td>Encryption</td>
<td>SHA-256</td>
</tr>
<tr>
<td>Beacon frequency</td>
<td>0.1 s</td>
<td>$P_{\text{noise}}$</td>
<td>-99 dBm</td>
</tr>
<tr>
<td>$\lambda_V$</td>
<td>[50, 200] nodes / km$^2$</td>
<td>$R$</td>
<td>250 m</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>3</td>
<td>$\beta_1$</td>
<td>8 dB</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Krauss</td>
<td>$d_{\text{min}}$</td>
<td>12 m</td>
</tr>
<tr>
<td>Average velocity</td>
<td>40 km/hr</td>
<td>$d_{\text{hop}}$</td>
<td>100 m</td>
</tr>
<tr>
<td>$L$</td>
<td>756 bytes</td>
<td>$W$</td>
<td>32</td>
</tr>
<tr>
<td>$T_{\text{RTS}}$</td>
<td>53 $\mu$s</td>
<td>$T_{\text{CTS}}$</td>
<td>37 $\mu$s</td>
</tr>
<tr>
<td>$T_{\text{DIFS}}$</td>
<td>58 $\mu$s</td>
<td>$T_{\text{SIFS}}$</td>
<td>32 $\mu$s</td>
</tr>
<tr>
<td>$T_{\text{ACK}}$</td>
<td>37 $\mu$s</td>
<td>$T_{\text{slot}}$</td>
<td>13 $\mu$s</td>
</tr>
<tr>
<td>$T_{\text{eye}}$</td>
<td>3332.11 $\mu$s</td>
<td>$DR$</td>
<td>6 Mbps</td>
</tr>
</tbody>
</table>

Table 3.4: Simulation parameters for PoQF message dissemination.

### 3.4.2 Security

Figure 3.6 shows $FV$ with respect to $n_{mn}$ at different $p_m$ and $n_{th}$. Two different values of $p_m$ are chosen to show the results at both low ($p_m < 0.5$) and high ($p_m > 0.5$) densities of malicious mining nodes presented in the network. As shown in Figure 3.6 (a), $FV$ with $p_m = 0.3$ is lower than $FV$ with $p_m = 0.7$ when $n_{th} = 3$, i.e., $n_{th} \leq \mu_m$. It shows that a low $n_{th}$ is suitable only for low $p_m$ when honest nodes are in majority. On the contrary, as shown in Figure 3.6 (b), $FV$ with $p_m = 0.3$ is higher than $FV$ with $p_m = 0.7$ when $n_{th} = n_{mn}$, i.e., $n_{th} > \mu_m$. This is because when $n_{th} = n_{mn}$ both malicious and honest mining nodes are unable to finalize consensus within the maximum allowable latency of 1s and the message is marked as false. With $p_m = 0.3$, the probability of false message occurrence is lower than that of true message occurrence and it is hard to collect $n_{th} = n_{mn}$ honest votes to validate a true message. In this case, $FV_{\text{UB}} \approx 1 - p_m$ depicts the worst-case scenario of maximum probability of true message generation which will be marked as false. With $p_m = 0.7$, the probability of true message occurrence is lower than that of false message occurrence. $FV$ does not occur when both honest and malicious nodes are unable
to collect votes for a false message. It only occurs when a true message is not validated. As shown in Figure 3.6 (b), $FV^{LB} \approx 1 - p_m$ with $p_m = 0.7$, depicts the percentage of true messages which are not validated by PoQF.

\[ FV \] with respect to $n_{mn}$.

Figure 3.6: $FV$ with respect to $n_{mn}$. 

(a) $n_{th} = 3$

(b) $n_{th} = n_{mn}$
(a) $CC_r = 100, TC = 0.5, n_{\text{hop}}^{\text{max}} = 10$

(b) $CC_r = 200, TC = 0.5, n_{\text{hop}}^{\text{max}} = 10$

(c) $CC_r = 200, TC = 0.1, n_{\text{hop}}^{\text{max}} = 6$

Figure 3.7: Setting up $CC_r$, $TC$ and $n_{\text{hop}}^{\text{max}}$ for collusion resistant incentive distribution.
Figure 3.7 shows the impact of parameters: $CC_r$, $TC$ and $n_{\text{hop}}^{\text{max}}$, on the collusion resistance feature of incentive distribution mechanism. According to Theorem 3.2, the incentive distribution mechanism is collusion resistant if $n_\text{h}TC \leq CC_{mn} \leq \frac{n_mCC_r}{n_{\text{hop}}^{\text{max}}(n_\text{h}-n_m)}$ and $p_m \leq 0.5$. As $B_i$ follows Binomial distribution, it can be assumed that $n_m \approx \mu_m$ and $n_\text{h} \approx \mu_\text{h}$. Based on this assumption, Figure 3.7 (a) and (b) show that the incentive distribution mechanism cannot be collusion resistant for every $n_{mn}, p_m$ under the fixed $CC_r, TC$ and $n_{\text{hop}}^{\text{max}}$. However, in Figure 3.7 (c), when $CC_r = 200, TC = 0.1, n_{\text{hop}}^{\text{max}} = 6$, $n_\text{h}TC \leq \frac{n_mCC_r}{n_{\text{hop}}^{\text{max}}(n_\text{h}-n_m)}$ is satisfied for every $n_{mn} \in (10,40)$ and $p_m \in (0,0.5)$. Therefore, for a collusion resistant incentive distribution mechanism, it is required that the combination of $CC_r$, $TC$ and $n_{\text{hop}}^{\text{max}}$ are adjusted with varying $n_{mn}$ and $p_m$, such that it is possible to choose $CC_{mn}$ within the boundaries defined by Theorem 3.2. Apart from the security reason, a low $n_{\text{hop}}^{\text{max}}$ is also favourable for successful message delivery, as the failure of multi-hop connectivity in VANETs increases with the number of hops.

Figure 3.8: Setting up $\omega_1$ and $\omega_2$ for collusion resistant incentive distribution.
Figure 3.8 shows the combined effect of $Rep_{ORG}$ and range of values of $\omega_1$ and $\omega_2$ resulting in different $U_i$ and $U_{RLY}$ according to incentive distribution mechanism 2 defined in Section 3.2.5 (c). In simulation, $CC = 10/Rep_{ORG}$ is arbitrarily set. Higher $Rep_{ORG}$ leads to lower amount of $CC$, thereby resulting in less $U_i$ and $U_{RLY}$. However, it must be ensured that blockchain-enabled message dissemination results in profit gain of all nodes, despite of deduction of $TC$. As shown in Figure 3.8, $\omega_1 > 0.3$ and $\omega_2 > 0.6$ would always result in positive $U_i$ and $U_{RLY}$, irrespective of the value $Rep_{ORG}$. A positive expected utility motivates players to play honestly and discourages malicious actions, therefore resulting in a secure incentive distribution mechanism.

3.4.3 Latency and Throughput

Figure 3.9 shows $T_{delay}$ of successful message validation with respect to $n_{mn}$ at different values of $p_m$ with $n_{th} = \mu_m + 1$. It can be seen that $T_{delay}$ increases with $p_m$ because of more frequent generation of microblocks by malicious nodes. Figure 3.9 (c) and (d) show that $T_{delay}^{UB}$ exceeds the maximum allowable latency requirement of 1s [3] when $p_m \geq 0.6$ and $n_{mn} \geq 30$. At $p_m = 0.8$ and $n_{mn} \geq 35$, the mining nodes are unable to finalize consensus within 1s.

Figure 3.10 shows $\lambda_{KB}$ of PoQF consensus at various $n_{mn}$ and $p_m$. The highest $\lambda_{KB}$ achieved is 11 keyblock/s at $p_m = 0.1$ and $n_{mn} = 15$ and the lowest is 0.9 keyblock/s at $p_m \geq 0.8$ and $n_{mn} = 35$, which means that at higher $n_{mn}$ and $p_m$, PoQF with $\lambda_{KB} < 1$ keyblock/s may not be able to generate block within the limit of maximum allowable latency of 1s. This shows that the proposed blockchain exhibits better performance specifically at lower values of $n_{mn}$ and $p_m$. $\lambda_{KB}$ can be improved by offloading computations required for encrypting a keyblock to a nearby edge computing server which have high computation power, thereby reducing $T_{eyp}$.
Figure 3.9: $T_{delay}$ with respect to $n_{mn}$. 

(a) $p_m = 0.2$ 

(b) $p_m = 0.4$ 

(c) $p_m = 0.6$ 

(d) $p_m = 0.8$
Figure 3.10: $\lambda_{KB}$ with respect to $p_m$.

Figure 3.11: Average success rate with respect to speed.

Figure 3.11 shows the success rate of transmitting a true message under different maximum speeds of a mining node. For $n_{hop} > 1$, PoQF consensus is only used for RLY selection since the message is already validated at $n_{hop} = 1$, and therefore, the transmission success rate is independent of $p_m$. It shows that the success rate is falling with increasing speed, specifically for small $n_{mn}$. This is because a small $n_{mn}$ depicts a low traffic density, so the nodes are likely to attain their maximum speeds and may lose connectivity before finalizing a consensus to select a RLY. In order to speed up consensus, a possible solution is that the RSUs reduce $n_{th}$ when $n_{hop} > 1$. At $n_{hop} > 1$, the consensus only depends on the highest $QF_i$ and does not require message validation. Since the mining node $i$ sends its microblock earlier than the mining node $j$ with $QF_j < QF_i$, it is not necessary for the mining node $i$ to
wait until $QF_j$ is received. It is noted that in case of an incident or traffic jam, a high speed is not likely to be attained in the affected area and therefore, it is not recommended to reduce $n_{th}$ at $n_{hop} = 1$, as it may result in large $FV$.

Figure 3.12 displays $a_{\tau_i}$ and $b_{\tau_i}$, as the lower and upper limits of random $\tau_i$ generated by a mining node $i$. It shows that $a_{\tau_i}$ and $b_{\tau_i}$ reduce with an increasing $QF_i$ and therefore $\tau_i$ leads to less waiting time for potential RLYs. Due to homogeneous distribution of nodes, $n_{neigh}$ is the same for every node in a network. Therefore, deliberately reducing $\tau_i$ by node $i$ is bounded by the lower limit of $a_{\tau_i}$, which is known to every node in a network. Such attempt can be easily detected and reported to CA.

3.4.4 Comparison of PoQF with other Consensus Algorithms

Figure 3.13 compares the performance of PoQF with PoET and PoS. In Figure 3.13 (a), $FV$ of PoQF is compared with PoET and PoS at different values of $p_m$ and $n_{mn}$, while $n_{th}$ is fixed at $\mu_m + 1$, as it results in low $FV$ for all values of $p_m$. PoET is implemented such that each node generates a random number between 0 to 1s to determine its waiting time for collecting microblocks. It shows that $FV$ of PoQF and PoET are closing to each other at low $p_m$. For high values of $p_m$, an honest node $i$ with the highest $QF_i$ is unable to collect sufficient microblocks from honest mining nodes within a random waiting time of PoET and therefore its $FV$ rises with $p_m$ at a higher rate than PoQF. In the reputation based PoS, a node is considered honest if its reputation exceeds a certain threshold. A reputation value is randomly assigned to each node on a scale of 0 to 100, thus the probability of reputation
falling below a threshold of 50 is defined as $p_m$. PoS is implemented such that a malicious RLY only forwards the message from a malicious sender and an honest RLY only forwards the message from an honest sender. On an average, PoQF reduces $FV$ by 11% and 15%, as compared to PoS and PoET, respectively.

Figure 3.13: Comparison of PoQF with PoS and PoET at $n_{th} = \mu_m + 1$. 
Figure 3.13 (b) compares the number of forks created by PoQF, PoS and PoET consensus. A blockchain consensus should be able to avoid creation of forks in order to control discrepancies. In PoQF, node \( i \) with the highest \( QF_i \) is most likely to announce its microblock prior to other mining nodes. In this way, node \( j \) with \( QF_j < QF_i \) cannot become \( RLY \), if votes of both nodes are the same. This is how creation of forks is reduced in the proposed consensus. PoS is implemented by selecting \( RLY \) on the basis of the highest reputation which is randomly generated from 0 to 100 in the simulation. A fork appears when two nodes with same reputation simultaneously become \( RLY \). In PoET, the time to announce microblock is not dependent upon \( QF \). Therefore, node \( j \) with lower \( QF_j \) becomes \( RLY \) before receiving a microblock from node \( i \) even though \( QF_i > QF_j \). In that case, a fork appears if both node \( i \) and node \( j \) generate keyblocks. It is noted that the number of forks in PoS is equal or lower than PoQF when \( n_{mn} \leq 20 \). Due to unreliable nature of vehicle connectivity, there remains a possibility of fork occurrence when an announced microblock by node \( i \) is not received by node \( j \). It usually happens when mining nodes are in distance and beyond the transmission range of each other. This is why a low node density, ultimately leading to low \( n_{mn} \), may result in a higher or equal number of forks created by PoQF as compared to PoS.

Figure 3.13 (c) compares \( T_{delay} \) of successful message validation consumed by PoQF, PoET and PoS. By using PoET, the mining node \( i \) is allowed to announce its microblock at a random time irrespective of its \( QF_i \). On an average, \( T_{delay} \) of PoET is 68ms higher than PoQF. However, the difference is larger at lower \( p_m \). Since \( \tau_i \) is independent of \( QF_i \) in PoET, node \( j \) with lower \( QF_j \) may announce its microblock earlier than node \( i \) with \( QF_i > QF_j \) and node \( i \) might have to wait longer. This waiting time is reduced in PoQF. However, with large \( p_m \), an honest mining node \( i \) with the highest \( QF_i \) has to wait longer in PoQF for receiving \( n_{th} \) honest microblocks. It is because the frequency of malicious microblocks generation is increased with a large \( p_m \). Hence the \( T_{delay} \) difference between PoQF and PoET becomes smaller. \( T_{delay} \) of PoS is independent of \( p_m \) and increases with \( n_{mn} \). It is the smallest because it only consumes time in \( RLY \) selection, while the voting time is eliminated by the reputation-based message validation.
<table>
<thead>
<tr>
<th>Consensus</th>
<th>Advantages</th>
<th>Limitations</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBFT [37]</td>
<td>• High throughput</td>
<td>• Low adversary control</td>
<td>• Message validation</td>
</tr>
<tr>
<td></td>
<td>• Fair to all nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoW [110]</td>
<td>• High security</td>
<td>• High power consumption</td>
<td>• Prevents security attacks</td>
</tr>
<tr>
<td></td>
<td>• Low throughput</td>
<td>• Low power consumption</td>
<td></td>
</tr>
<tr>
<td>PoS [7]</td>
<td>• High throughput</td>
<td>• Demotivating for nodes with low stakes</td>
<td>• Trust based data sharing</td>
</tr>
<tr>
<td></td>
<td>• Low power consumption</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint PoW and PoS [92]</td>
<td>• Faster than PoW</td>
<td>• High power consumption</td>
<td>• Reputation management</td>
</tr>
<tr>
<td></td>
<td>• Fairer than PoS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoQF</td>
<td>• High adversary control</td>
<td>• Involves probabilistic predictions</td>
<td>• Message validation</td>
</tr>
<tr>
<td></td>
<td>• High throughput</td>
<td></td>
<td>• Relay node selection</td>
</tr>
<tr>
<td></td>
<td>• Fair to all nodes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.5: Comparison of PoQF with other consensus algorithms for blockchain-enabled VANETs.

Table 3.5 compares performance of various consensus algorithms including PoQF and their potential applications in VANETs. It shows that PoQF can be employed for both message validation and relay selection.

3.5 Summary

In this chapter, a blockchain based PoQF consensus algorithm is proposed for message dissemination in vehicular networks. The theoretical performance of the proposed consensus is evaluated by deriving bounds on failure and latency in message validation, throughput of block generation and asymptotic latency, security and communication complexity. Moreover, two incentive distribution mechanisms have been presented to promote positive cooperation and discourage malicious behaviour of nodes and analysed using game theory. The proposed PoQF offers both high throughput and high adversary control.
control and serves the dual purpose of message validation and relay selection. However, it relies on regular beacon message in which private information of nodes including its position and speed are shared.
Chapter 4 – Blockchain-enabled Federated Learning (FL) for Message Dissemination in Vehicular Networks

In this chapter, a decentralised message dissemination solution using a hierarchical blockchain based FL process is proposed, which is aimed to reduce latency and enhance privacy of a voting-based message dissemination. Section 4.1 introduces FL, the motivation of integrating FL with blockchain and challenges of FL in vehicular networks. Section 4.2 describes the proposed solution of blockchain-enabled FL for multi-hop relay selection. An incentive distribution mechanism to reward nodes participating in FL and message dissemination is also designed and analysed using Stackelberg game model. Both theoretical and simulated performance evaluation of the proposed approach are presented in section 4.3 and section 4.4 respectively.

4.1 Introduction

As discussed in Chapter 2, multi-hop relaying is one of the approaches in V2V communications to successfully deliver a message over a wide area. Optimal relay selection mechanisms result in better coverage, more reliable connectivity, and less communication overhead [127]. A voting based relay selection algorithm, such as the one described in Chapter 3 takes considerable time in waiting for votes, which increases with rising percentage of malicious nodes in the network. Furthermore, it relies heavily on probabilistic predictions and is prone to cheating attempts. For example, a malicious node may share false information in beacon messages. Alternatively, various intelligent relay selection schemes depending on a node’s distance from predecessor, moving direction, speed and propagation loss in environment have been proposed using fuzzy logic [30] or machine learning algorithms [32]. Existing literature shows improved packet delivery ratio by machine learning algorithms in multi-hop V2V communications [128]. However, AI methods require huge processing power and are often not suitable for a fully distributed architecture [33]. In a traditional centralised architecture of machine learning, the data collected by mobile nodes is uploaded and processed in a cloud-based server to produce inference models [129]. With potentially large number of nodes in VANETs, where real-time decisions must be made within a restricted time, a cloud-centric approach is unable to
offer acceptable latency and scalability. Also, a centralised architecture requires full connectivity which is challenging for VANETs. FL is an intermediary solution combining the advantages of both distributed learning and central aggregation.

### 4.1.1 Federated Learning (FL)

It is a distributed machine learning approach, in which nodes collect their own private data and train their individual machine learning or deep learning models, called local models, as shown in Figure 4.1. They do not share their collected data but only send self-trained local models to an aggregator. The aggregator combines all local models and produces a global model. The nodes further train the global model individually to create updated local models and submit them to aggregator. These steps are repeated in multiple iterations until a desired accuracy of global model is achieved [130]. FL is considered as a feasible solution for safety and time critical applications involving vehicles [54]. However, due to heterogeneity of data in vehicular networks, the performance of a global model may vary significantly across different nodes, environments, and traffic situations [131].

![Figure 4.1: The FL process.](image)

### 4.1.2 FL meets Blockchain

Despite offering a distributed approach, FL still relies on a central aggregator. Furthermore, it needs a sustainable incentive distribution mechanism to reward cooperative nodes based on their contributions and prevent adversary attacks. For example, a malicious node may deliberately modify data used for creating a local model, thereby causing poisoning attack
or a selfish node may not cooperate in data collection resulting in inaccurate weights of a local model.

<table>
<thead>
<tr>
<th>FL issues</th>
<th>Blockchain based solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires central aggregator</td>
<td>Independent of third party</td>
</tr>
<tr>
<td>Needs incentive strategy to motivate cooperative nodes</td>
<td>Manages cryptocurrency-based incentives</td>
</tr>
<tr>
<td>Requires adversary control</td>
<td>Provides security by smart contract</td>
</tr>
</tbody>
</table>

Table 4.1: FL issues and blockchain solutions.

Blockchain can be used with FL to provide a decentralised solution, manage incentives and ensure security and privacy in a trustworthy manner. Due to its decentralised nature, blockchain complements both FL and VANETs. Furthermore, smart contracts can enforce security by ensuring that a set of rules is followed [68]. The process of transaction verification in blockchain can also be utilised to validate local models in FL [55]. Table 4.1 summarises the current issues of FL and corresponding solutions provided by blockchain.

4.1.3 FL in Vehicular Networks

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Limitations</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance / link quality based predictions [27]</td>
<td>Assumptions/ rules are not adaptable to</td>
<td>Local models trained with different networks and the global model can cater network changes</td>
</tr>
<tr>
<td>Fuzzy logic [30]</td>
<td>network changes</td>
<td></td>
</tr>
<tr>
<td>Machine learning [32] - [33]</td>
<td>Huge data have to be managed centrally</td>
<td>Distributed learning and decentralised storage</td>
</tr>
<tr>
<td>Any scheme without incentives [26]</td>
<td>RLYs may act selfishly</td>
<td>Blockchain incentives for motivation</td>
</tr>
</tbody>
</table>

Table 4.2: Multi-hop relay selection limitations and solutions offered by blockchain-based FL.
FL is suggested as a promising technique to securely train intelligent models across smart cars [54] and Unmanned Aerial Vehicles (UAVs) [133]. It has the feature of reducing network latency by dividing training task among network edges. In C-V2X communications, FL is proposed to reduce failure probability by intelligently offloading high computation tasks to nearby base stations [134]. Resource allocation and sharing in C-V2X by FL among nodes has promised better coverage and Quality-of-Service (QoS) in [135]. FL and fog-assisted V2X is presented in [136] to improve driving experience of vehicles by providing user-end services, for example, car sharing, intelligent parking allocation, infotainment, and e-commerce applications. In [137], FL is used to tackle energy transfer issues of electric vehicles at charging stations and has resulted in improved accuracy of energy demand prediction. FL assisted blockchain is proposed in [138] to adjust block arrival rate to reduce communication latency and consensus delay among nodes. Applications of FL in vehicular networks are summarised in [139] and most of the recent applications focus on resource management, performance optimization in computing tasks and user-end services. However, FL can also be promising in message delivery and RLY selection. Table 4.2 lists the limitations in existing multi-hop relay selection schemes and solutions offered by blockchain-based FL.

4.1.4 Contributions

a) Modes of FL

FL can be carried out in two modes: synchronous and asynchronous, as shown in Figure 4.2. In synchronous FL, nodes are given a time limit in which they submit their local models. A global model is formed after a time epoch. In asynchronous FL, nodes can send their models at their own convenient time. A global model is updated each time a new model is received [135]. Synchronous FL promises more accuracy and convergence [54]. However, a node may not be able to upload its local model in a fixed time epoch due to high mobility in a vehicular network, as shown in Figure 4.3 (a). On the other hand, in asynchronous FL in VANET, it is possible that a node loses connection with the aggregator and is unable to receive every update in global model. It may continue training its local model on the basis of an outdated global model, thereby leading to waste of its resources, as shown in Figure 4.3 (b). To increase accuracy and reduce packet loss, a modified
synchronous FL can be used in VANET in which the aggregator waits for a certain number of local models to be received instead of a time epoch. This chapter employs modified synchronous FL and waits for a certain number of local models before aggregation.

Figure 4.2: Modes of FL.

Figure 4.3: Modes of FL in VANET.
b) **Consensus**

Selection of an appropriate consensus is one of the challenges of blockchain-enabled FL in VANETs. In [138], the drawback of additional delay due to blockchain management in FL process of vehicular networks is discussed. A possible reason of delay is PoW being used by nodes to record and validate local models. To reduce the delay, DPoS is used as a consensus of blockchain-supported FL in [140]. The proposed approach is faster but decentralization is compromised because only RSUs can generate blocks. In addition, a consensus algorithm in blockchain-enabled FL requires not only time-efficiency but also security against malicious nodes. As a solution, this chapter uses a machine learning based algorithm embedded in smart contract to replace consensus and ensure security against malicious local models.

### 4.2 Blockchain-based FL for Message Dissemination

![Blockchain-based FL and Message Dissemination](image)

Figure 4.4: Blockchain-based FL and Message Dissemination.

As illustrated in Figure 4.4, the proposed solution consists of a blockchain-based FL process and a solution for multi-hop relay selection. The FL process is aimed to form a global model which is later used as a consensus to select $RLY$ when an incident message is originated by a node. The vehicles act as nodes participating in FL by training local models...
and RSU performs the role of aggregator. Overall, the proposed approach consists of two major parts:

- **Blockchain-supported FL** where nodes complete an FL process to create a global model for RLY selection.
- **Proof-of-Federated-Learning (PoFL) based blockchain-enabled message dissemination**, where global model produced in the above solution is run by nodes to find their eligibility of becoming RLY.

### 4.2.1 Blockchain-supported FL

#### a) Preliminary elements

- **Hello packet by designated node**: A CA appoints some designated nodes to regularly originate a Hello packet and share their position to initiate dataset collection by nodes participating in FL. Only the designated nodes are allowed to originate Hello packets. The motivation behind designated nodes is two-fold: first is because they are trusted by CA to honestly send their actual position without any malicious change and second is because the identities of designated nodes are already shared with other nodes, so Hello packet from any other identity is not recognized by the network. Designated vehicles can either be representatives of Central Authority or selected from the existing network based on their reputation.

- **Dataset**: In the proposed solution, dataset collected by node $i$ includes multi-hop relay selection parameters collected as data samples to train local model. After forwarding a Hello packet, dataset collected by node $i$ consists of the following parameters: $d_{i,s}$; distance from sender $s$ (designated node or previous RLY), $\text{dir}_{i,s}$; moving direction (either towards or away from sender $s$), $v_i$; speed at the time of forwarding message, $n_{\text{hop}}$; hop number, $\lambda_v$; traffic density within its transmission range and $n_A$; number of acknowledgments received as the score of relaying. $\lambda_v$ in dataset can be pre-specified by CA or estimated by counting average number of nodes sending beacon messages per meter [118]. The process of dataset collection is explained in detail later in this chapter.
b) **FL components supported by blockchain**

Table 4.3 defines components involved in an FL and supporting elements for each corresponding component in a blockchain-enabled process.

<table>
<thead>
<tr>
<th>Component of an FL process</th>
<th>Blockchain Support</th>
</tr>
</thead>
</table>
| **Adversary:** The following adversary threats are considered in an FL process:  
  a) Malicious nodes: They may deliberately change or inject false data so that local model is not trained accurately. This phenomenon is also known as poisoning attack [132].  
  b) Selfish nodes: They may not send acknowledgment messages despite receiving forwarded messages. Therefore, $n_A$ cannot be recorded correctly during dataset collection, leading to an inaccurate local model produced by node $i$. | **Security Check:** It is a machine learning algorithm embedded in smart contract of FL blockchain to detect adversary before a local model is uploaded as a block by node $i$. |
| **Local Model:** Each node $i$ participating in FL trains a Deep Neural Network based local model consisting of 7 hidden layers and 256 neurons in each layer. | **Microblock:** A local model is stored in FL blockchain as a microblock after undergoing a security check. A microblock is added in parallel to other microblocks, all containing hash of previous keyblock. |
| **Global Model:** It is an aggregated model of $n_{FL}$ local models, where $n_{FL}$ also refers to the number of nodes participating in FL. The global model is consolidated by RSU in the proposed solution. | **Keyblock:** A global model is stored in FL blockchain by RSU in the form of a keyblock, containing hashes of previous $n_{FL}$ microblocks. |

Table 4.3: FL components and corresponding blockchain support.
c) The proposed solution

Algorithm 4.1: Actions by node $i$ to complete blockchain-support FL

**Input:** Hello packet

**Output:** Global model

1. while $n_{hop} \leq n_{hop}^{max}$ do
2.   Generate random waiting time
3.   while time elapsed $\leq$ random waiting time do
4.     if Forwarded Hello packet received at $n_{hop}$ then
5.       Break
6.     end if
7.   end while
8.   if Forwarded Hello packet not received at $n_{hop}$ then
9.     Forward Hello packet
10.    Count acknowledgement packets into $n_A$
11.   Record in $d_{i,s}, v_i, dir_{i,s}, \lambda_{i}, n_A$ in dataset
12.   Break
13.  Else
14.   $n_{hop} = n_{hop} + 1$
15.  end if
16. end while
17. if data size $== s_i$ then
18.   Train local model
19. else
20.   Go to 1
21. end if
22. while $k \leq k_{max}$ do
23.   Upload local model through smart contract
24.   Receive updated global model
25.   Re-train local model
26.   $k = k + 1$
27. end while
Figure 4.5: The proposed stages of blockchain-supported FL.

The proposed blockchain-supported FL consists of three stages, as shown in Figure 4.5, and explained below. The solution is also described in Algorithm 4.1.

- **Stage 1 - Dataset Collection and Local Model Training:** In this stage, nodes collect dataset for training. Upon receiving a Hello packet from a designated node, a node \( i \) which aims to collect dataset, generates a random waiting time. When the waiting time is complete, it forwards Hello packet with its encrypted identity. The reason behind a random waiting time is to prevent multiple nodes from transmitting at the same time and avoid packet collision. The nodes which receive the forwarded Hello packet for the first time share their acknowledgment. An acknowledgment packet contains encrypted identity of node \( i \), so that it can collect dataset. A node \( j \), which participates in FL, will broadcast the received Hello packet again after a random waiting time. This process continues up to a specified number of hops, \( n_{\text{hop}}^{\text{max}} \).

- **Stage 2 - Security Check and FL Blockchain Update:** A node \( i \) shares its local model with the network by adding it into FL blockchain as a microblock. It is added after passing a security check performed by the smart contract embedded in FL blockchain. The proposed security check employs a machine learning algorithm called Isolation Forest [141] to detect anomaly in a local model caused by adversary. Isolation Forest is used because it only requires a small number of samples for training. A true sample is provided by the CA for its initial training. Later, it can be used in a fully unsupervised manner to detect anomaly. Moreover,
it is computationally efficient and has low memory requirement [142]. The security check is used in three ways. Firstly, the security check on dataset is conducted by finding anomalies in dataset of each node. Secondly, the security check performs anomaly detection on weights of local models. If a malicious node $i$ deliberately changes its dataset for training its local model but shares a true dataset in smart contract, the adversary attack will be detected by anomaly detection on weights. Thirdly, the security check on both dataset and weights is performed. If local models successfully pass the security check, they are added in FL blockchain in the form of parallel microblocks. The microblock announcement is broadcasted by node $i$ and the receiving nodes will then update their copy of FL blockchain. Nodes can exchange new microblock updates with their neighbours regularly.

- **Stage 3 - Global Model Aggregation by RSU**: Whenever a node $i$ finds an RSU available in its transmission range, it shares its updated copy of FL blockchain. When $n_{FL}$ microblocks are received by RSU in FL blockchain, it aggregates local models into a global model and uploads it into a keyblock. All stages are repeated at each iteration. The goal is to repeat the process up to $k_{max}$ iterations for minimizing global loss function $LF(w^k_G)$, which is defined as

$$LF(w^k_G) = \frac{1}{n_{FL}} \sum_{i=1}^{n_{FL}} LF(w^k_i),$$

where $w^k_G$ are weights of global model and $LF(w^k_i)$ is the loss function of local model by node $i$ and $w^k_i$ are its corresponding weights at $k^{th}$ iteration. Neural networks commonly use Mean Squared Error (MSE) as the loss function [54]. The value of $k_{max}$ is adjusted by CA to achieve the minimum possible or desired $LF(w^{k_{max}}_G)$ [130].

### 4.2.2 PoFL based Blockchain-enabled Message Dissemination

**a) Components**

- **PoFL**: It is the consensus to elect RLY for forwarding an incident message, initiated by ORG. It uses the global model contained in the latest keyblock of FL blockchain, described in above solution. It is run by a smart contract of message blockchain.
• Message Blockchain: It contains history of incident messages. When a RLY is elected by PoFL, it adds a block in the message blockchain containing the forwarded incident message. The block also contains time, location and encrypted identity of the RLY which adds the block. The motivation behind message blockchain is to record forwarded incident messages as immutable blocks and avoid discrepancies during incentive distribution among RLYs.

b) The proposed solution

Algorithm 4.2: PoFL based message dissemination

<table>
<thead>
<tr>
<th>Input: Incident message, global model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: New block announcement in message blockchain</td>
</tr>
</tbody>
</table>

1. while \( n_{hop} \leq n_{hop}^{max} \) do  
2. Compute score from global model  
3. timer expiry limit = \( 1/score \)  
4. while time elapsed \( \leq \) timer expiry limit do  
5. if New block announced then  
6. Break  
7. end if  
8. end while  
9. if New block not announced then  
10. Announce block  
11. Break  
12. Else  
13. \( n_{hop} = n_{hop} + 1 \)  
14. end if  
15. end while

When an incident message is initiated by ORG, all receiving nodes attempt to become the RLY by competing through PoFL consensus. Each node \( i \) runs PoFL consensus to find its score of being RLY. PoFL is aimed to assign the highest score to the most appropriate RLY. The smart contract starts a timer whose length is inversely proportional to the score of node \( i \). As shown in Figure 4.6, a block is added in the message blockchain and a block
announcement with the forwarded incident message is initiated by node \( i \) if its timer first expires. In this case, node \( i \) is assigned as \( RLY \) at \( n_{hop} = 1 \). All other nodes continue to compete for becoming \( RLY \) at further hops until the message is forwarded up to \( n_{hop}^{max} \) number of hops. The solution is also described in Algorithm 4.2. The relationship between two solutions is illustrated in Figure 4.7.

Figure 4.6: Flowchart of actions by node \( i \) according to PoFL based blockchain-enabled message dissemination.
Figure 4.7: The flow of steps in blockchain-supported FL and PoFL based message dissemination.

c) Privacy of FL based message dissemination

<table>
<thead>
<tr>
<th>Approach</th>
<th>Position</th>
<th>Speed</th>
<th>Other parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance / link quality based predications [26] - [27]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Fuzzy logic [30]</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Deep learning [33]</td>
<td>✓</td>
<td>✓</td>
<td>Transmission power</td>
</tr>
<tr>
<td>PoQF (Chapter 3)</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>PoFL</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 4.4: Parameter-sharing required from neighbour nodes.

Table 4.4 lists the parameters required to be shared by neighbour nodes in various multi-hop relay selection approaches. The position, speed and heading direction of moving nodes are regularly shared in VANETs using beacon messages and thus create a threat to privacy [143]. The proposed approach does not require such information from all neighbour nodes and can therefore be considered as a privacy-preserving solution. The position and direction of only sender is required for dataset collection in blockchain based FL and for score calculation of relaying in PoFL based message dissemination. However, identities of nodes are kept anonymous using encryption. Disclosure of identities through brute force
attack is a possibility but it will not be very effective for the attacker. Due to high time complexity of brute force attack [144], the position, speed and direction of a node will be changed until its identity is disclosed. In case of high probability of brute force attack, a private blockchain with only trusted nodes or timely refreshing of cryptographic identities is recommended [145].

4.3 Theoretical Analysis

4.3.1 Training Capacity of FL

This analysis is aimed to compare the capacity of blockchain to complete one FL iteration in a given amount of time with the same process without blockchain. FL without blockchain is referred to as a centralised approach in which each node $i$ submits its local model directly to RSU instead of uploading it into FL blockchain. As the convergence performance of FL improves with increasing number of local models [146], FL via blockchain is expected to achieve greater accuracy, provided the number of uploaded local models are higher as compared to the process carried out without blockchain within the same time period.

a) FL with blockchain

Let $TS$ be a time slot in which a node $i$ is required to upload its local model as microblock in FL blockchain, after it has completed training and passed its local model through security check. Let $\lambda_{MB}$ be the microblock arrival rate at RSU or throughput in microblocks/s. If microblock arrival is modelled using Poisson distribution as defined in [75], the expected number of nodes able to upload their local models via FL blockchain in $TS$ can be given as [120]

$$E(n_B) = \sum_{l=1}^{\lambda_{MB}TS} l e^{-\lambda_{MB}TS} \frac{(\lambda_{MB}TS)^l}{l!}.$$  (4.2)

b) FL without blockchain

If moving nodes are required to upload their models directly to RSU without blockchain, then it is necessary that either RSU is in their transmission range or they are able to reach
towards RSU within $T_S$. Consider a general and dynamic movement of nodes, the position of nodes (vehicles) on road follows Poisson distribution, $\lambda_V$ nodes/m$^2$ is assumed as the density of nodes on a two-dimensional road segment with no separation of lanes [117], the expected number of nodes with RSU in their transmission range is

$$E(n_V) = \sum_{l=1}^{\lambda_V\pi R^2} l e^{-\lambda_V\pi R^2} \frac{\left(\lambda_V\pi R^2\right)^l}{l!}, \quad (4.3)$$

where transmission range is assumed as a uniform circle with radius $R$. Similarly, the expected number of moving nodes with RSU not currently in their transmission range but can travel to reach RSU within $T_S$ is

$$E(n'_V) = \sum_{l=1}^{\frac{T_S}{\mu_d - R\mu_v}} l e^{-\frac{T_S}{\mu_d - R\mu_v}} \frac{\left(\frac{T_S}{\mu_d - R\mu_v}\right)^l}{l!}, \quad (4.4)$$

where $\mu_d > R$ is the mean distance of those nodes from RSU which do not have RSU within their transmission range and $\mu_v$ is their average speed. Therefore, the expected number of nodes able to upload their local models to RSU without blockchain during $T_S$ is

$$E(n_{WB}) = E(n_V) + E(n'_V). \quad (4.5)$$

### 4.3.2 Incentive Distribution Mechanism and its Analysis

An incentive distribution mechanism to reward nodes contributing in FL and message dissemination is formulated. The feasibility of incentive strategy is analysed by investigating behaviour of RLYs and nodes participating in FL based on their expected utilities through Stackelberg game model.

#### a) Stackelberg game formulation

The Stackelberg game model consists of three types of players: ORG, RLY participating in message dissemination and node $i$ participating in FL. For each incident message initiated by ORG, there are $n_{RLY}$ number of RLYs which forward the incident message and $n_{FL}$ nodes in the network which train their local models during blockchain based FL. The proposed incentive distribution mechanism is formulated as a two-stage Stackelberg game.
First, at stage 1, ORG pays reward to RLYs for forwarding message. At stage 2, RLYs pay reward to \(n_{FL}\) nodes for participating in FL to form a global model of RLY selection. Since, both FL and message dissemination processes are blockchain-based, the contribution of players is stored as immutable timestamped blocks and cannot be altered for cheating. The transactions of incentives are also processed in the form blockchain based virtual currency automatically through smart contracts. As shown in Table 4.5, the rewards to \(n_{FL}\) nodes are paid in proportion to the sizes of dataset they have used in training their local models. Assume that the dataset sizes of \(n_{FL}\) nodes are \(s = \{s_1, s_2, ..., s_{n_{FL}}\}\). The utility of each node \(i\) participating in FL process is

\[
U_i(s_i, I) = n_{FL}I s_i - Cost(s_i),
\]

where \(I\) denotes incentive which is constant for every node \(i\) and \(Cost(s_i)\) is the computational cost of training a local model on dataset of size \(s_i\) and is modelled as a quadratic function, i.e.,

\[
Cost(s_i) = \rho_i s_i^2,
\]

where \(\rho_i > 0\) denotes cost co-efficient of node \(i\) [147]. The utility of each RLY is

\[
U_{RLY}(s, I) = CClog(1 + I) - \sum_{i=1}^{N} I \cdot s_i.
\]

where \(CClog(1 + I)\) is paid by ORG for forwarding the incident message. Here \(CC > 0\) and can be assumed as a compensation amount paid to RLYs present in an area affected by an incident or traffic jam caused by ORG.

<table>
<thead>
<tr>
<th>Player</th>
<th>Gains</th>
<th>Pays</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORG</td>
<td>None</td>
<td>(n_{RLY}CClog(1 + I)) to RLYs</td>
</tr>
<tr>
<td>RLY</td>
<td>(CClog(1 + I)) from ORG</td>
<td>(\sum_{i=1}^{N} I s_i) to (n_{FL}) nodes</td>
</tr>
<tr>
<td>Node (i)</td>
<td>(n_{RLY}I s_i) from RLYs</td>
<td>(\rho_i s_i^2) to train local model</td>
</tr>
</tbody>
</table>

Table 4.5: Reward gained and payment made by players.
b) **Stackelberg game analysis**

The proposed incentive distribution mechanism assumes information symmetry, i.e., all players have complete information about $s_i$.

- **Definition 4.1**: Assume that $s^*_i$ is the optimal data size for each node $i$ and $I^*$ is the optimal incentive amount per data size paid by each RLY to node $i$, then $(s^*_i, I^*)$ is the Nash equilibrium point which satisfies the following conditions

$$U_i(s^*_i, I^*) \geq U_i(s_i, I^*),$$

(4.9)

$$U_{RLY} (s^*_i, I^*) \geq U_{RLY} (s_i^*, I).$$

(4.10)

- **Theorem 4.1**: There exists a Nash equilibrium point for a node $i$ with $U_i$ defined in (4.9).

*Proof:* For a fixed $I^*$, $U_i$ is

$$U_i(s_i, I^*) = N_{RLY} \cdot I^* \cdot s_i - \rho_i s_i^2.$$  

(4.11)

The second order derivative of (4.11) is

$$\frac{\partial^2 U_i(s_i, I^*)}{\partial s_i^2} = -2\rho_i.$$  

(4.12)

Since $\rho_i > 0$, the second-order derivative of $U_i$ is negative and $U_i(s_i, I^*)$ is a strictly concave function, which proves the existence of Nash equilibrium.  

$\blacksquare$

- **Theorem 4.2**: There exists a Nash equilibrium point for RLY with $U_{RLY}$ defined in (4.10).

*Proof:* The second order derivative of (4.10) is

$$\frac{\partial^2 U_{RLY}(s, I)}{\partial I^2} = -\frac{CC}{(1 + I)^2}.$$  

(4.13)
Since $CC > 0$ and $(1 + I)^2 > 0$, the second-order derivative of $U_{RLY}$ is negative and $U_{RLY}(s, I)$ is a strictly concave function, which proves the existence of Nash equilibrium.

Based on Theorem 4.1 and Theorem 4.2, it can be stated that the unique Stackelberg Nash equilibrium point of the proposed model exists. CA is responsible to choose values of $I$ and $CC$ such that Nash equilibrium points for all $n_{FL}$ nodes and $n_{RLY}$ RLYs become their best response strategies (i.e., $U_i > 0$ and $U_{RLY} > 0$) and all players are willing to cooperate in the proposed game.

If permissioned blockchain is used, complete information may not be visible to every player and the incentive distribution mechanism will be information asymmetric. In this case, players may predict information using probabilistic assumption [147] or through a machine learning method [148]. Let $s = \{s_1, s_2, \ldots, s_Z\}$ be the $n_z$ dataset sizes used by nodes and $p_z$ be the probability that a node $i$ uses $s_z$. (4.10) can be modified as

$$U_{RLY}(s, I) = CC \log \gamma(1 + I) - \sum_{z=1}^{n_z} p_z n_{FL} l s_z,$$

(4.14)

Similar to Theorem 4.2, the existence of Nash Equilibrium point can be proved for $U_{RLY}$ defined in (4.14).

**4.3.3 Asymptotic Complexities**

<table>
<thead>
<tr>
<th>Consensus</th>
<th>Latency</th>
<th>Communication</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoS</td>
<td>$\Omega(\kappa)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>PoQF</td>
<td>$\kappa O(1)$</td>
<td>$O(n_{mn})$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>PoFL</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>Blockchain-enabled FL</td>
<td>$\Omega(\kappa)$</td>
<td>$\Omega(n_{FL})$</td>
<td>$n_{FL} \Omega(\kappa)$</td>
</tr>
</tbody>
</table>

Table 4.6: Comparison of asymptotic complexities.

Table 4.6 compares the asymptotic latency, communication and computation complexities of PoS, PoQF, PoFL and the blockchain-based FL process which is used to produce a global model for PoFL. $\kappa$ denotes consensus parameter and is unique to each algorithm.
which is the smallest dataset size in blockchain-based FL. The blockchain-based FL process has the highest computation complexity, as its computations depend on the size of dataset and number of vehicles submitting local models. Also, its communication complexity is proportional to the number of nodes participating in FL and its latency depends on the time of training a local model, which is proportional to the dataset size. However, PoFL, resulted from a blockchain-based FL process, outperforms PoS and PoQF because its latency, communication and computation complexities are independent of $n_{FL}$ and $k$. It is therefore a highly scalable solution once the FL process is completed.

4.4 Simulated Performance Analysis

4.4.1 Simulation Setup

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>300 s</td>
<td>Protocol</td>
<td>IEEE 802.11p</td>
</tr>
<tr>
<td>Size of area</td>
<td>2.5 km $\times$ 2.5 km</td>
<td>Encryption</td>
<td>SHA-256</td>
</tr>
<tr>
<td>$DR$</td>
<td>6 Mbps</td>
<td>$s_i$</td>
<td>8000</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Krauss</td>
<td>Loss function</td>
<td>MSE</td>
</tr>
<tr>
<td>Number of RSUs</td>
<td>1</td>
<td>$R$</td>
<td>250 m</td>
</tr>
<tr>
<td>Number of nodes (vehicles)</td>
<td>100, 200, 300</td>
<td>$n_{hop}^{max}$</td>
<td>6</td>
</tr>
<tr>
<td>$k_{max}$</td>
<td>100, 110</td>
<td>$\mu_v$</td>
<td>50 km/hr</td>
</tr>
</tbody>
</table>

Table 4.7: Simulation parameters for blockchain-based FL and message dissemination.

The proposed solutions are simulated using OMNeT++, Python and SUMO\(^2\). Python is employed for carrying out FL using Tensorflow library of machine learning. Python can be embedded into C++ by writing an extension module. Since OMNeT++ is a modular C++ based network simulator, it supports dynamic loading of Python script at run time. Appendix C shows code snippet of OMNeT++ integrating Python. The simulation parameters used are listed in Table 4.7.

\(^2\) Source code is available at [https://zenodo.org/record/5575863#.YW2BmOhKiUk](https://zenodo.org/record/5575863#.YW2BmOhKiUk).
4.4.2 Loss Function of Global Model

(a) No security check

(b) Security check on dataset

(c) Security check on weights

(d) Security check on both dataset and weights

Figure 4.8: Loss (MSE) of global model with 50% adversary.
Figure 4.8 shows the loss (MSE) of global model with respect to iteration index $k$, in presence of 50% adversary among the nodes participating in FL, consisting of equal percentage of malicious and selfish nodes. In all cases, the loss converges to its lowest possible value until 100 iterations. However, this convergence is achieved in less number of iterations with 300 nodes as compared to 100 nodes, which means that the maximum accuracy of a global model can be attained faster with greater number of nodes. Figure 4.8 (a) shows the loss when no security check is implemented. The convergence rate is slower without security check and takes more iterations than those with security checks, as shown in Figure 4.8 (b) - (d). It indicates that the increased computation required to implement security check can be compensated with less iterations processed to attain maximum accuracy. Table 4.8 shows loss of global model after 100 iterations with respect to number of nodes participating in FL without any adversary or security check. As shown in Table 4.8, the loss is inversely proportional to both dataset size and number of nodes.

<table>
<thead>
<tr>
<th>$s_i$</th>
<th>$n_{FL} = 100$</th>
<th>$n_{FL} = 200$</th>
<th>$n_{FL} = 300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>0.19643</td>
<td>0.18724</td>
<td>0.16541</td>
</tr>
<tr>
<td>5000</td>
<td>0.17251</td>
<td>0.17021</td>
<td>0.16313</td>
</tr>
<tr>
<td>8000</td>
<td>0.15297</td>
<td>0.15101</td>
<td>0.15085</td>
</tr>
</tbody>
</table>

Table 4.8: Loss (MSE) of global model after 100 iterations.

Figure 4.10 and Figure 4.10 show the loss of global model after 100 iterations of FL in presence of malicious and selfish nodes respectively. The global loss function is the highest if no security check is employed in smart contract of FL blockchain. Security check on weights results in less loss as compared to security check on dataset in presence of malicious nodes and vice-versa in case of selfish nodes. It is because selfish nodes affect only $n_A$ in dataset by not sending acknowledgments and such discrepancy is easily detected if security check is applied on dataset only. On the other hand, malicious nodes can change all parameters in dataset and therefore it is not easy to detect anomaly on such dataset. It shows that security check on weights is more suitable to prevent poisoning attack caused by malicious nodes and security check on dataset is more appropriate to reduce the effect of selfish behaviour. Nevertheless, malicious nodes may submit a true dataset for security check and upload inaccurate local models using a false dataset. Therefore, in this case, only
security check on weights can prevent adversary caused by malicious nodes. The loss function is minimum if security check on both dataset and weights is used and is suitable for both malicious and selfish nodes. As a trade-off, increased computation is required to run security check twice.

![Figure 4.9: Loss (MSE) of global model after 100 iterations with malicious nodes.](image)
4.4.3 Evaluation of Training Capacity of FL with and without Blockchain

Figure 4.11 displays the average number of nodes over 100 simulation runs which uploaded their local models during $T_S$, with and without blockchain at various $\lambda_V$. The simulation results are matched with expected values derived in (4.2) and (4.5), confirming theoretical analysis. $\lambda_{MB}$ and $\mu_d$ change with varying $\lambda_V$ and are listed in Table 4.9. Figure 4.11 shows that blockchain based approach results in average 18 more nodes uploading their local models within same $T_S$ compared with the centralised solution in submitting local models.
directly to RSU without blockchain. This is because a copy of FL blockchain is possessed by each node. A local model by a node $i$ can be entered into FL blockchain without depending upon RSU. Subsequently, RSU is able to receive an updated FL blockchain by another node $j$, containing local models of both node $i$ and node $j$. Without blockchain, a node $i$ must travel towards RSU within $T_S$ to directly share its local model. In this case, one RSU or small $T_S$ may not be sufficient for receiving local models from large number of nodes. Also, as shown in Table 4.9, the loss of global model decreases with rising $n_{FL}$. It can be concluded that FL blockchain can achieve desired accuracy of a global model faster than FL carried out without blockchain, because FL blockchain enables collection of local models from a greater number of nodes within the same time limit.

<table>
<thead>
<tr>
<th>$\lambda_V$ (nodes/m$^2$)</th>
<th>16</th>
<th>32</th>
<th>48</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{MB}$ (microblocks/s)</td>
<td>2.01</td>
<td>1.99</td>
<td>0.98</td>
</tr>
<tr>
<td>$\mu_d$ (m)</td>
<td>344</td>
<td>298</td>
<td>276</td>
</tr>
</tbody>
</table>

Table 4.9: $\lambda_{MB}$ and $\mu_d$ with respect to $\lambda_V$.

![Figure 4.11: Number of nodes uploading local model in $T_S$.](image-url)
4.4.4 Evaluation of Incentive Distribution Mechanism

(a) $U_i$

(b) $U_{RLY}$ at $n_{FL} = 200, CC = 0.9 \times 10^7$.

(c) $U_{RLY}$ at $n_{FL} = 200, CC = 1.8 \times 10^7$.

Figure 4.12: Utility of players in Stackelberg Game with equilibrium points (*).
Figure 4.12 proves Definition 4.1 at equilibrium points at various values of $\rho_i$, $CC$ and $n_{FL}$. Figure 4.12 (a) shows the utility of node $i$, $U_i$, participating in a blockchain based FL. As shown in Figure 4.12 (a), for a given $I^*$, there exists only one $s_i^*$ which results in maximum $U_i$. Figure 4.12 (b) and (c) show $U_{RLY}$ at $n_{FL} = 200$ with varying values of $s_i$ and $CC$. In each case, an equilibrium point exists where $U_{RLY}$ is maximum for a given $I^*$. A CA can select the value of $I^*$, which gives both maximum $U_i$ and $U_{RLY}$. As shown in Figure 4.12 (b), $U_{RLY} \leq 0$ for certain values of $I$, which will motivate $RLY$s to become selfish. An appropriate value of $CC$ can be selected to make $U_{RLY} > 0$ for every value of $I$, as shown in Figure 4.12 (c). A machine learning model can be used to predict the optimum values of $CC$ and $I$, according to $s_i$, which result in best response strategy of $RLY$s. This model can be embedded into smart contract of message blockchain to automate reward distribution independently without CA.

4.4.5 Comparison of PoFL with other Solutions

Figure 4.13: Message delivery ratio with 300 nodes.
Figure 4.13 shows message delivery ratio among 300 nodes at varying percentages of malicious nodes in the network as a result of 100 simulation runs. Results are compared with PoQF described in Chapter 3. PoFL outperforms voting based relay selection method and PoQF when security check is applied. However, with low percentage of malicious nodes and maximum speed of 55 km/hr, PoQF results in better message delivery ratio than PoFL when security check on weights is not applied. Nevertheless, PoFL with security check on both dataset and weights always outperforms voting based relay selection and PoQF by an average of 25% and 8.2% increase in message delivery ratio respectively.

Figure 4.14 shows the average time delay per hop in completing PoFL, PoQF and PoS consensus. PoS is simulated such that it selects $RLY$ on the basis of reputation of node. A random reputation value following uniform distribution, ranging from 0 to 100 is assigned to each node. The average time delay per hop of PoQF rises with increasing number of nodes and percentage of malicious nodes in the network. This is because PoQF waits for a threshold number of votes to determine $RLY$ and the optimum threshold increases with rising number of total nodes and malicious nodes percentage. Time delay of PoS rises with increasing number of nodes due to more time required in accessing large amount of reputation values but it is independent of percentage of malicious nodes. PoFL is run by each node simultaneously and therefore its time delay is independent of both number of nodes and malicious percentage. On an average, PoFL is 65.2% faster than PoQF in relay selection and is more suitable for time-critical emergency situations. As a trade-off, PoQF only involves Quality Factor calculations but PoFL is based on a computationally...
expensive FL process with multiple iterations. Compared to PoS, PoFL is 15.74% faster when there are 300 nodes but 18.93% slower when there are 100 nodes. This is because PoS consumes time only in accessing the blockchain to find reputation of nodes. The access time increases when there is a large number of nodes registered in a blockchain network. Although PoS with 100 nodes outperforms both PoQF and PoFL, this faster consensus for block verification and addition cannot be run independently for appropriate relay selection, unlike PoQF and PoFL.

4.5 Summary

In this chapter, a decentralised FL based message dissemination, governed by blockchain is introduced. The theoretical and practical performances of uploading local models using blockchain are compared with a centralised approach without blockchain. The proposed FL with blockchain can be considered as a more efficient approach since it results in greater number of uploaded local models within a given time as compared to a solution without blockchain. Smart contract based security checks are proposed to detect adversary, which result in lower MSE in less number of iterations achieved by global model than FL without security check. An incentive distribution mechanism for blockchain based FL is also proposed and analysed using Stackelberg game to determine optimal data size and incentive which result in the best response strategy of players.

The future work can include FL based message validation process before relay selection. To achieve a light-weight solution and reduced computation cost, alternative consensus algorithms can be used to prevent malicious attacks instead of machine learning method embedded in smart contract. For example, a local model may be considered credible if reputation of the trainer exceeds a certain threshold.
Chapter 5 — Blockchain-enabled FD-NOMA based Vehicular Network with Physical Layer Security

In this chapter, a blockchain based vehicular network using FD-NOMA is analysed with physical layer security ensuring a transmission’s reliability and secrecy rate. The chapter is divided into three main sections. Background and motivation are discussed in section 5.1. In section 5.2, an FD-NOMA based vehicular network is theoretically analysed for security and privacy of the proposed network against eavesdropping and jamming. We derive the upper bound of blockchain throughput in terms of physical layer characteristics, i.e., SINR and secrecy rate. In section 0, Monte Carlo simulations are presented to validate the theoretical analysis.

5.1 Overview

5.1.1 Introduction

There are several attacks on security and privacy which can affect the performance of vehicular networks. These attacks can be classified on the basis of layer used by an attacker in a communications protocol stack [149]. From physical layer's perspective, a vehicular network may suffer from the following attacks [10], [150]:

a) **Jamming Attack**

It is caused when an attacker creates interference to disrupt the communications between sender and legitimate receiver. It affects the *availability* attribute of security in a vehicular network.

b) **Eavesdropping Attack**

It is caused when an eavesdropper intercepts the transmission between sender and legitimate receiver. It affects the *confidentiality* attribute of privacy in a vehicular network.

Security and privacy are usually managed at the upper layers of a communications protocol stack by using various techniques including blockchain and key based encryption [38]. Blockchain has been proven resilient against security attacks, such as application layer attack of repudiation [151], network layer attacks of Denial of Service and Sybil [152]. A
permissioned blockchain not only ensures security but also privacy, by allowing only authorized nodes to join and communicate in a network. However, physical layer attacks can severely affect performance of a blockchain-enabled vehicular network. For example, jamming can disrupt block announcement, thereby reducing throughput of blockchain. PLS is another effective approach to protect privacy and maintain secrecy against eavesdropping [153] but may also result in a decreased blockchain throughput.

5.1.2 Motivation

Motivated by the latest advances in blockchain and the importance of security and privacy in vehicular communications, we study the effects of physical layer attacks on blockchain-enabled vehicular network. Additionally, due to increasing number of connected nodes, heterogeneous environment and requirement of high communication rates, FD-NOMA based V2X communications scenario is considered [154]. An FD-NOMA model fulfils the requirements of various QoS and multiple communication rates in V2X systems [155]. It also addresses the issue of low latency in existing Orthogonal Frequency Division Multiple Access based 5G technologies by simultaneous transmission and retrieval of data [156]. It is particularly suggested for V2X applications, e.g., navigation and emergency message dissemination [154]. Its roadmap for V2X based services has already been prepared by technical organizations, such as 3rd Generation Partnership Project [157]. Furthermore, NOMA based techniques can be used to provide security against jamming attack by nullifying co-channel interferences via Successive Interference Cancellation (SIC) [154].

The effect of interference on blockchain throughput is also studied in [120]. SINR is a physical layer parameter and a function of distances between nodes in wireless communications. It can be severely degraded by attackers causing jamming and interference in signal transmission and therefore result in a reduced blockchain throughput. However, FD-NOMA can significantly improve SINR by SIC. Therefore, the performance of FD-NOMA for a secure blockchain based V2X system is worth investigating. Performance of FD-NOMA based V2X systems is analysed in [154] assuming both Rayleigh and Rician channel models. The analysis does not take into account the traffic density, speeds or distances between moving nodes which greatly affect the reliability of signal transmission in V2X systems.
Furthermore, some of the security and privacy requirements of V2X systems may not be met by blockchain alone. For example, if a blockchain uses voting consensus, the votes must also be encrypted in presence of an eavesdropper, which increases computation and communication overheads. Moreover, if all nodes broadcast their votes at the same time, the reliability of a transmission is severely reduced by interference. Therefore, an integrated approach considering both physical layer aspects, e.g., SINR and secrecy rate, and application layer schemes, e.g., blockchain, can be utilized to provide robust security in vehicular networks. However, the feasibility analysis of an integrated approach is required before its practical implementation.

5.2 Analysis of Blockchain-enabled FD-NOMA based Vehicular Network

5.2.1 System Model

Figure 5.1: The system model.

Figure 5.1 illustrates a V2X network consisting of mobile and stationary nodes including vehicles, pedestrian, RSU, base station etc. forming an FD-NOMA based system. The
nodes can be categorised into one of the following: sender, legitimate receiver, interferer and eavesdropper. We assume urban and crowded environment is assumed; hence all communication channels are modelled by Rayleigh fading [158]. The channel matrix from $n_s$ sender nodes to $n_r$ receiver nodes in FD-NOMA based decentralised V2X systems is defined in [154] as

$$ H = \begin{bmatrix} h_{1,1} & \cdots & h_{n_s,1} \\ \vdots & \ddots & \vdots \\ h_{1,n_r} & \cdots & h_{n_s,n_r} \end{bmatrix}, $$

(5.1)

where $h_{i,j} = \sqrt{g_{i,j} d_{i,j}^{-\alpha}}$ is the channel coefficient between node $i$ and node $j$ and $g_{i,j}$ is the channel gain following Rayleigh fading [38]. Assume that all channels are uncorrelated and have increasing order of channel coefficients, i.e., $|h_{1,j}| \leq |h_{2,j}| \leq \ldots \leq |h_{i,j}| \leq \ldots \leq |h_{n_s,j}| \forall i \in [1, n_s], j \in [1, n_r]$. In this case, co-channel interference of $j^{th}$ node is from $(j + 1)^{th}$ to $n_s^{th}$ node. Other co-channel interferences are nullified by SIC feature of NOMA [154].

Due to high mobility of nodes in a vehicular network, we take into account the uncertainty of nodes' positions. Therefore, $d_{i,j}$ is assumed as a random variable following exponential distribution. Exponential distribution has been shown as a suitable approximation to model traffic flow condition [159]. The PDF of $d_{i,j}$ is

$$ f(d_{i,j}) = \frac{1}{\overline{d}_{i,j}} e^{-\frac{d_{i,j}}{\overline{d}_{i,j}}}, $$

(5.2)

where $\overline{d}_{i,j}$ is the average distance between node $i$ and node $j$. The traffic density in nodes/km can be defined as $1000/\overline{d}_{i,j}$ [160].

The performance of a blockchain-enabled wireless network is characterized by two important parameters: data rate and blockchain throughput. Data rate is defined as the amount of transmitted data in a unit time for a network, usually measured in bits per second (bps). Blockchain throughput, $\lambda_B$, is the number of blocks validated and generated in a unit
time. It is measured in blocks per second (blocks/s). The relationship of data rate $DR$ with blockchain throughput $\lambda_B$ and block length $L$ is defined in [120] as

$$DR \geq \lambda_B \cdot L. \quad (5.3)$$

### 5.2.2 SINR

When a signal is received by node $j$ from node $i$, the instantaneous SINR is defined in [154] as

$$SINR_{i,j} = \frac{P_i |h_{i,j}|^2}{\sum_{l=j+1}^{n_s} P_l |h_{i,l}|^2 + \eta P_j + P_{noise}}, \quad (5.4)$$

where $\eta P_j$ is the self-interference by FD up-link, $\eta \in [0,1]$ is the coefficient of self-interference, $P_{noise}$ is the noise power of Additive White Gaussian Noise (AWGN), $P_l$ and $P_i$ are the power of signal transmitted by node $i$ and interference node $l$ respectively. The received signal is subjected to only co-channel interference from neighbours of node $j$ after SIC.

In urban and crowded environment, the PDF of $SINR_{i,j}$ is given in [154], [161] as

$$f(SINR_{i,j}) = \frac{1}{SINR_{i,j}} e^{-\frac{SINR_{i,j}}{\overline{SINR}_{i,j}}}, \quad (5.5)$$

where $\overline{SINR}_{i,j}$ is the average SINR. It is noted that $SINR_{i,j}$ can be modelled as a random variable following exponential distribution. As shown in (5.4), it depends on $h_{i,j}$, which is a function of $d_{i,j}^{-\alpha}$. Since $d_{i,j}$ is also an exponential variable, as shown in (5.2), Lemma 5.1 and Theorem 5.1 are presented to derive bounds of $\overline{SINR}_{i,j}$ as a function of $d_{i,j}$ and $n_{itf}^j$, i.e., the number of interference nodes to receiver $j$.

- **Lemma 5.1**: $E\left(\frac{1}{d_{i,j}^{-\alpha}}\right) = \overline{d_{i,j}}^{-\alpha} \left(\Gamma(\alpha + 1, d_{min}/\overline{d_{i,j}}) - \Gamma(\alpha + 1, d_{max}/\overline{d_{i,j}})\right)$, where $d_{min}$ is the minimum $d_{i,j}$ and $d_{max}$ is the maximum $d_{i,j}$ up to which a signal can be transmitted.
Proof: Since

\[ E(1/X) = \int_{-\infty}^{\infty} x^{-1} f(x) \, dx, \quad (5.6) \]

let \( X = d_{i,j}^{-\alpha} \), then its CDF is

\[ F_X(x) = Pr(X \leq x) = Pr\left(d_{i,j}^{-\alpha} \leq x\right) \]

\[ = Pr\left(d_{i,j} > x^{-1/\alpha}\right) = 1 - F_{d_{i,j}}(x^{-1/\alpha}). \quad (5.7) \]

PDF of \( X \) can be obtained by taking derivative of (5.7), i.e.,

\[ f_X(x) = \frac{1}{\alpha} x^{-\alpha - 1} f_{d_{i,j}}(x^{-1/\alpha}) = \frac{1}{d_{i,j} \alpha} x^{-\alpha - 1} e^{\frac{-1}{d_{i,j}}} \quad (5.8) \]

Combining (5.6) and (5.8) gives

\[ E(1/X) = \frac{1}{d_{i,j} \alpha} \int_{d_{\text{min}}^{-\alpha}}^{d_{\text{max}}^{-\alpha}} x^{-\alpha - 2} e^{\frac{x^{-1/\alpha}}{d_{i,j}}} \, dx, \quad (5.9) \]

Solving (5.9) gives \( E\left(\frac{1}{d_{i,j}^{-\alpha}}\right) = \frac{d_{i,j}^{-\alpha}}{\Gamma(\alpha + 1, d_{\text{min}}/d_{i,j})} - \Gamma(\alpha + 1, d_{\text{max}}/d_{i,j}) \). \( \blacksquare \)

- **Theorem 5.1:**

\[ \frac{1}{n_{i,j}^l E\left(d_{i,j}^{-\alpha}\right) + P'} E\left(\frac{1}{d_{i,j}^{-\alpha}}\right) \leq \frac{d_{i,j}^{-\alpha}}{n_{i,j}^l E\left(d_{i,j}^{-\alpha}\right) + P'}, \quad \text{where } E\left(\frac{1}{d_{i,j}^{-\alpha}}\right) \]

is defined in Lemma 5.1, \( P' = \frac{nP_l + P_{\text{noise}}}{P_g} \) and \( P = P_l = P_t, g = g_{i,j} = g_{t,l} \forall l \in n_s \), without loss of generality.

Proof: Assuming \( P = P_l = P_t \) and \( g = g_{i,j} = g_{t,l} \forall l \in n_s \), (5.4) can be rewritten as

\[ SINR_{i,j} = \frac{d_{i,j}^{-\alpha}}{n_{i,j}^l d_{i,j}^{-\alpha} + P'} \quad (5.10) \]

and therefore,

\[ \frac{1}{SINR_{i,j}} = \frac{n_{i,j}^l d_{i,j}^{-\alpha}}{d_{i,j}^{-\alpha} + P'} \quad (5.11) \]
\[
E \left( \frac{1}{\text{SINR}_{ij}} \right) = n_{tf}^j E \left( \frac{d_{i,j}^{-\alpha}}{d_{i,j}^{-\alpha}} \right) + P'E \left( \frac{1}{d_{i,j}^{-\alpha}} \right).
\]  
(5.12)

Let \( E \left( \frac{d_{i,j}^{-\alpha}}{d_{i,j}^{-\alpha}} \right) = E \left( \frac{Y}{X} \right) = E(Y) \cdot E \left( \frac{1}{X} \right) \) where \( X = d_{i,j}^{-\alpha} \) and \( Y = d_{i,j}^{-\alpha} \). Since both \( d_{i,j} \) and \( d_{i,j} \) represent distance between nodes, \( E(Y) = E(X) = \frac{1}{d_{i,j}^{-\alpha}} \int_{d_{\text{min}}^{-\alpha}}^{d_{\text{max}}^{-\alpha}} x^{-\frac{1}{\alpha}} e^{-\frac{x}{d_{ij}}} dx \).

According to Jensen’s inequality [162], \( \frac{1}{E(\text{SINR}_{ij})} \leq E \left( \frac{1}{\text{SINR}_{ij}} \right) \), which follows that

\[
\text{SINR}_{ij} \geq \frac{1}{E \left( \frac{1}{\text{SINR}_{ij}} \right)}.
\]

Since \( \text{SINR}_{ij} \) is directly proportional to \( d_{i,j}^{-\alpha} \), \( \text{SINR}_{ij} \leq \)

\[
\frac{d_{i,j}^{-\alpha}}{n_{tf}^j d_{\text{max}}^{-\alpha} + P'}
\]

\[\square\]

For a reliable message transmission and successful block generation, it is necessary that \( \text{SINR}_{ij} \) exceeds a certain threshold. The probability that \( \text{SINR}_{ij} \) exceeds a threshold \( \beta_1 \) can be given by its CDF, i.e.,

\[
F_{\text{SINR}_{ij}}(\beta_1) = \int_{0}^{\beta_1} f(\text{SINR}_{ij}) d\text{SINR}_{ij} = \int_{0}^{\beta_1} \frac{1}{\text{SINR}_{ij}} e^{\frac{\text{SINR}_{ij}}{\beta_1}} d\text{SINR}_{ij}
\]

\[= 1 - e^{-\frac{\beta_1}{\text{SINR}_{ij}}},\]

(5.13)

and

\[
\Pr(\text{SINR}_{ij} \geq \beta_1) = 1 - F_{\text{SINR}_{ij}}(\beta_1) = e^{-\frac{\beta_1}{\text{SINR}_{ij}}}.
\]

(5.14)

Also, \( e^{-\frac{\beta_1}{\text{SINR}_{ij}}} \leq \Pr(\text{SINR}_{ij} \geq \beta_1) \leq e^{-\frac{\beta_1}{\text{SINR}_{ij}} L} \). In case of a jamming attack caused by interference, a block can only be generated if it is transmitted successfully to at least one legitimate receiver. Therefore, (5.3) is modified as follows

\[
DR \geq \frac{\lambda_B}{\Pr(\text{SINR}_{ij} \geq \beta_1) L} \cdot L.
\]

(5.15)
5.2.3 Secrecy Rate

If a signal from node \( i \) is sent to a legitimate receiver node \( j \) but a node \( k \) attempts to receive the signal as an eavesdropper, as shown in Figure 5.1, the secrecy rate is defined in [163] as

\[
C_{i,j} = [C_j - C_k]^+,
\]

where \( C_j = \log_2(1 + \text{SINR}_{i,j}) \), \( C_k = \sum_{k=1}^{n_e} \log_2(1 + \text{SINR}_{i,k}) \), \( n_e \) is the number of eavesdroppers present in the communication range of node \( i \) and \([.]^+\) denotes \( \max(.,0) \).

Using logarithmic property, i.e., \( \log(a) + \log(b) = \log(ab) \), \( C_k \) can also be represented as \( C_k = \log_2 \phi \), where \( \phi = (1 + \text{SINR}_{i,1})(1 + \text{SINR}_{i,2}) \ldots (1 + \text{SINR}_{i,n_e}) \).

PLS ensures that a message is transmitted when secrecy rate is greater than a certain threshold \( \beta_2 \). Theorem 5.2 defines the probability of maintaining confidentiality through PLS.

**Theorem 5.2:**

\[
Pr(C_{i,j} \geq \beta_2) = \begin{cases} 
\frac{1-2\beta_2}{2\beta_2 S\text{INR}_{i,j} + S\text{INR}_{i,j}} S\text{INR}_{i,j}^2 e^{S\text{INR}_{i,j}} \frac{1}{u} E_1 \left(\frac{v_1 v_2}{u}\right) e^{-\frac{v_1 v_2}{u}}, & n_e = 1, \\
\frac{-2\beta_2}{2\beta_2 S\text{INR}_{i,j} + S\text{INR}_{i,j}} S\text{INR}_{i,j}^2 e^{S\text{INR}_{i,j}} \frac{1}{u} E_1 \left(\frac{v_1 v_2}{u}\right) e^{-\frac{v_1 v_2}{u}}, & n_e = 2,
\end{cases}
\]

where \( u = 2\beta_2 S\text{INR}_{i,j} \cdot S\text{INR}_{i,k=1} \cdot S\text{INR}_{i,k=2} \), \( v_1 = S\text{INR}_{i,j} + 2\beta_2 S\text{INR}_{i,k=2} \), \( v_2 = v_1 + 2\beta_2 + 1 S\text{INR}_{i,k=1} \) and \( E_1(a) = \int_a^{\infty} e^{-z} \frac{dz}{z} \) is exponential integral.

\[
Pr(C_{i,j} \geq \beta_2) \approx 0 \text{ for } n_e > 2.
\]

**Proof:** Since \( C_{i,j} \) is a function of \( \text{SINR}_{i,j} \) and \( \text{SINR}_{i,k} \), \( Pr(C_{i,j} \geq \beta_2) \) is given in [161] as
\[ Pr(C_{i,j} \geq \beta_2) = \int_{\text{SINR}_{i,n_e}=0}^{\infty} \ldots \int_{\text{SINR}_{i,1}=0}^{\infty} f(\text{SINR}_{i,j}) f(\text{SINR}_{i,1}) \ldots \]
\[ \ldots f(\text{SINR}_{i,n_e}) d\text{SINR}_{i,j} d\text{SINR}_{i,1} \ldots d\text{SINR}_{i,n_e}. \tag{5.17} \]

When \( n_e = 1 \), (5.17) reduces to
\[ Pr(C_{i,j} \geq \beta_2) = \int_{0}^{\infty} y^{\frac{1}{2}} \phi^{-\frac{1}{2}} \int_{0}^{\infty} f(\text{SINR}_{i,j}) f(\text{SINR}_{i,1}) d\text{SINR}_{i,j} d\text{SINR}_{i,1}. \tag{5.18} \]

Using \( \int_{t}^{\infty} \int_{0}^{\tfrac{1}{a}} {\tfrac{z}{t}} e^{-\tfrac{z}{a}} dz = e^{-\tfrac{t}{a}} \), (5.18) becomes
\[ Pr(C_{i,j} \geq \beta_2) = \frac{1}{\text{SINR}_{i,1}} \int_{0}^{\infty} e^{-\frac{2\beta^2(1 + \text{SINR}_{i,1}) - 1}{\text{SINR}_{i,j}}} \frac{\text{SINR}_{i,1}}{\text{SINR}_{i,j}} d\text{SINR}_{i,1}, \tag{5.19} \]

which follows that
\[ Pr(C_{i,j} \geq \beta_2) = \frac{\text{SINR}_{i,j} e^{\text{SINR}_{i,j}}}{2\beta^2 \text{SINR}_{i,1} + \text{SINR}_{i,j}}. \tag{5.20} \]

For \( n_e = 2 \), the first two integrals of (5.17) can be solved similarly and it becomes
\[ Pr(C_{i,j} \geq \beta_2) \]
\[ = \frac{1}{\text{SINR}_{i,1}} \int_{0}^{\infty} e^{\text{SINR}_{i,j}} f(\text{SINR}_{i,2}) d\text{SINR}_{i,2} \]
\[ = \frac{\text{SINR}_{i,j} e^{\text{SINR}_{i,j}} E_1 \left( \frac{\nu_1 \nu_2}{u} \right) e^{-\frac{\nu_1 \nu_2}{u}}}{u}. \tag{5.21} \]

As (5.21) involves exponential integral, obtaining a closed form equation of \( Pr(C_{i,j} \leq \beta_2) \) for \( n_e > 2 \) is at least arduous, if not impossible [38], [154]. However, it can be seen that the resulting values of \( u, \nu_1 \) and \( \nu_2 \) after solving (5.17) for \( n_e > 2 \) will increase and will lead to \( Pr(C_{i,j} \geq \beta_2) \approx 0. \)
Theorem 5.2 shows that an FD-NOMA transmission without blockchain may not provide secrecy when \( n_e > 2 \), since \( \Pr(\mathcal{C}_{i,j} \geq \beta_2) \approx 0 \). Therefore, privacy preserving measures such as encryption schemes and blockchain are therefore essential in such cases, where confidentiality cannot be protected by PLS alone.

Using \( E_1(z)e^z \leq \log \left(1 + \frac{1}{z}\right) \) [164] and assuming \( e^{\frac{1-2\beta_2}{\text{SINR}_{i,j}^\text{UB}}} \approx 1 \), the upper bound of \( \Pr(\mathcal{C}_{i,j} \geq \beta_2) \) can be defined as

\[
\Pr(\mathcal{C}_{i,j} \geq \beta_2) \leq \begin{cases} 
\frac{\text{SINR}_{i,j}^\text{UB}}{2^{\beta_2} \text{SINR}_{t,1} + \text{SINR}_{i,j}^\text{UB}}, & n_e = 1, \\
\frac{\text{SINR}_{i,j}^\text{UB} \log \left(1 + \frac{u'}{v_1' v_2'}\right)}{u'}, & n_e = 2, \\
0, & \text{otherwise},
\end{cases}
\]

(5.22)

where \( u' = 2^{\beta_2} \cdot \text{SINR}_{t,1} \cdot \text{SINR}_{t,2} \), \( v_1' = \text{SINR}_{i,j}^\text{UB} + 2^{\beta_2} \text{SINR}_{t,1} \), \( v_2' = v_1' + 2^{\beta_2+1} \text{SINR}_{t,1} \).

In case of eavesdropping attack, the secrecy rate must be greater than \( \beta_2 \) for every receiver to protect confidentiality. Therefore, for protecting every legitimate receiver from eavesdropping attack on physical layer, (5.3) is modified as follows

\[
DR \geq \frac{\lambda_B}{\prod_{j=1}^{n_r} \Pr(\mathcal{C}_{i,j} \geq \beta_2)^\text{UB} \cdot L}.
\]

(5.23)

A sender can probabilistically estimate the presence and location of eavesdropper before transmission through solutions defined in [165] and [166].

5.2.4 Goodput

To analyse the impact of PLS combined with blockchain, we define the term goodput as \( DR \) times the ratio of number of blocks successfully and secretly added into the blockchain to the total number of block generation attempts, i.e.,
\[
\text{Goodput} = DR \cdot \frac{\text{No. of blocks added to blockchain}}{\text{Total No. of block generation attempts}}
\]  

(5.24)

where \( \text{Total No. of block generation attempts} = \text{No. of blocks added to blockchain} + \text{Number of blocks lost or eavesdropped} \).

5.3 Simulated Performance Analysis

In this section, simulation results are compared with theoretical analysis presented in Section 5.2. Monte Carlo simulations are conducted on MATLAB. The code for FD-NOMA implementation can be seen in Appendix D. The parameters used in simulations are listed in Table 5.1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>(10^5)</td>
<td>(\eta)</td>
<td>0.1</td>
</tr>
<tr>
<td>(P_i)</td>
<td>20 dBm</td>
<td>(P_{\text{noise}})</td>
<td>-104 dBm</td>
</tr>
<tr>
<td>(d_{\text{max}})</td>
<td>400 m</td>
<td>(d_{\text{min}})</td>
<td>10 m</td>
</tr>
<tr>
<td>(\beta_1)</td>
<td>-15 dB</td>
<td>(\beta_2)</td>
<td>0.3 bits/sec/Hz</td>
</tr>
<tr>
<td>(\lambda_B)</td>
<td>1, 50 blocks/s</td>
<td>(L)</td>
<td>756 bytes</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>3</td>
<td>(n_e)</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>(n_{\text{itf}})</td>
<td>[1, 5]</td>
<td>(n_r)</td>
<td>[2, 5]</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters used in simulation of FD-NOMA based vehicular network.

5.3.1 SINR

Figure 5.2 (a) shows \(\overline{\text{SINR}}_{i,j}\) varying with respect to \(\overline{d}_{i,j}\) at \(n_{\text{itf}}^1 = 1\) and \(n_{\text{itf}}^1 = 2\). The simulated \(\overline{\text{SINR}}_{i,j}\) lies within the bounds defined in Theorem 5.1, validating our analysis. It can be seen that \(\overline{\text{SINR}}_{i,j}\) falls with increasing \(\overline{d}_{i,j}\). The dependence of \(\overline{\text{SINR}}_{i,j}\) on \(\overline{d}_{i,j}\) is higher when \(n_{\text{itf}}^1 = 1\) as compared to \(n_{\text{itf}}^1 = 2\). It shows that \(\overline{\text{SINR}}_{i,j}\) can be enhanced by reducing \(\overline{d}_{i,j}\) only when interference is low. Figure 5.2 (b) shows \(Pr(\overline{\text{SINR}}_{i,j} \geq \beta_1)\) with respect to \(\overline{d}_{i,j}\). The theoretical result is computed using (5.14). In simulation, \(\overline{\text{SINR}}_{i,j} \geq \beta_1\) is counted as a successful transmission for each iteration. The percentage of successful
transmissions is plotted as a simulated result in Figure 5.2 (b). It can be seen in that $Pr(SINR_{i,j} \geq \beta_1)$ falls with increasing $d_{i,j}$ due to decreasing $SINR_{i,j}$. $Pr(SINR_{i,j} \geq \beta_1)$ is higher for less $n_{itf}^i$, which depicts the effect of interference. High interference is considered as a collusion of attackers to hinder successful transmission. This is why a high $Pr(SINR_{i,j} \geq \beta_1)$ is desired for a secure and reliable transmission.

![Figure 5.2: $SINR_{i,j}$ and $Pr(SINR_{i,j} \geq \beta_1)$](image)

Figure 5.2: $SINR_{i,j}$ and $Pr(SINR_{i,j} \geq \beta_1)$ with respect to $d_{i,j}$. 

(a) $SINR_{i,j}$

(b) $Pr(SINR_{i,j} \geq \beta_1)$
5.3.2 Secrecy Rate

Figure 5.3: $P_r(C_{i,j} \geq \beta_2)$ with respect to $\overline{d}_{i,j}$.

Figure 5.3 shows $Pr(C_{i,j} \geq \beta_2)$ with $n_e = 1$ and $n_e = 2$. The theoretical result and upper bound are plotted using Theorem 5.2 and (5.22) respectively. Simulations show the percentage of iterations which resulted in $C_{i,j} \geq \beta_2$. $Pr(C_{i,j} \geq \beta_2)$ reduces with increasing $n_e$. Figure 5.3 (b) shows that $Pr(C_{i,j} \geq \beta_2)$ is less than 50% when $n_e = 2$ and $n_{itf}^k = 1$, despite varying values of $n_{itf}^k$. It reflects that maintaining secrecy is extremely challenging.
with large number of eavesdroppers. Therefore, cryptographically protected blockchain is an effective solution to ensure confidentiality of a transmission in such case.

5.3.3 Percentage of Success Transmissions

Figure 5.4 shows percentage of success transmissions with respect to $d_{i,j}$ in presence of jammers and eavesdroppers. A success transmission is counted if $\text{SINR}_{i,j} \geq \beta_1$, with jammers only and also if $C_{i,j} \geq \beta_2$, when eavesdroppers are present. As shown in Figure 5.4, success rate is higher with jammers only than with eavesdroppers. Specifically, when $n_e = 2$, the success rate is below 50% for every $d_{i,j}$. As a PLS approach, a sender must estimate that $C_{i,j} \geq \beta_2$ to protect secrecy of a message. However, it is extremely challenging to attain $C_{i,j} \geq \beta_2$ in presence of large number of eavesdroppers. A cryptographically protected blockchain is an effective solution to ensure confidentiality of a transmission in such case. Therefore, a cross-layer approach combining both PLS and blockchain is promising to provide security against both attacks.

Figure 5.4: Success transmissions in presence of jammer and eavesdroppers, $n_{itf}^j = 2$, $n_{itf}^k = 1$. 

![Figure 5.4: Success transmissions in presence of jammer and eavesdroppers](image)
5.3.4 Minimum Allowable Data rate

Figure 5.5: Minimum allowable $DR$.

Figure 5.5 shows the lower bound of $DR$, i.e., minimum allowable $DR$ sufficient to support both PLS and blockchain as a combined solution against jamming and eavesdropping according to the relation derived in (5.15) and (5.23) respectively. As shown in Figure 5.5 (a), $DR^{LB} < 0.5$ Mbps for both $\lambda_B = 1$ block /s and $\lambda_B = 50$ block /s, which shows that the proposed approach does not require a very high data rate to provide security against
jamming attack. However, in (b), $DR^{LB}$ is rising with increase in $n_r$, $n_e$, $\lambda_B$ or $d_{i,j}$. Specifically, for certain values of $d_{i,j}$, when $n_r = 3$, it can be seen that $DR^{LB} > 100$ Mbps. Since IEEE 802.11p supports $DR$ ranging from 3 to 54 Mbps [167], it may not be feasible to implement a secure blockchain-enabled PLS solution when high blockchain throughput is required or large number of receivers are present. An integration of blockchain and PLS can be more effective with 5G or beyond 5G technologies which offer peak data rates in Gbps [168].

5.3.5 Goodput

![Figure 5.6: Goodput in presence of jammers and eavesdroppers.](image-url)

(a) With jammers

(b) With eavesdroppers, $n_{i,j}^f = 1$, $n_{i,j}^k = 2$. 

Figure 5.6: Goodput in presence of jammers and eavesdroppers.
Figure 5.6 shows comparison of goodput in a blockchain-based V2X system, with and without PLS, simulated in OMNeT++ at $DR = 27$ Mbps using IEEE 802.11p protocol. Figure 5.6 (a) depicts significant improvement in goodput by using PLS in presence of jammers. Goodput in presence of eavesdroppers is also improved, as shown in Figure 5.6 (b). On an average, there is an increase of 8.2 Mbps in goodput with PLS as compared to the same solution without PLS. However, the goodput falls with increase in $d_{i,j}$, $n_e$ or $n_r$ in all cases. With $n_e = 2$ and $n_r = 3$, no block is successfully and secretly added into the blockchain after a certain $\overline{d_{i,j}}$. It shows that there is no sufficient $DR$ available to support the integrated approach of blockchain and PLS. Nevertheless, strong cryptographic measures in blockchain can still protect confidentiality of eavesdropped blocks.

### 5.4 Summary

This chapter has analysed FD-NOMA based vehicular network which employs both PLS and blockchain to meet security and privacy requirements in presence of jammers and eavesdroppers. It can be concluded that integration of PLS and blockchain can provide better goodput against both jamming and eavesdropping attacks. However, it requires high $DR$ to support large number of legitimate receivers for protecting privacy in presence of eavesdroppers. DSRC based IEEE 802.11p communications may not provide sufficient $DR$ for feasible integration of PLS and blockchain. Therefore, 5G technologies are recommended for such applications. We plan to further evaluate and collect simulation results to thoroughly analyse the approach in practical scenarios.
Chapter 6 – Conclusions and Future Work

This final chapter concludes the thesis. It is divided into two main sections. Section 6.1 summarises the contributions of the research work. Section 6.2 highlights future research directions.

6.1 Contributions and Conclusions

The main focus of this thesis is to devise solutions for message dissemination in VANETs which can be secured by blockchain. The performance of the solutions is measured in terms of time delay in message validation and dissemination per hop, blockchain throughput, success rate in message delivery and tolerance (i.e., failure in message validation and loss in federated learning) with varying percentage of malicious nodes. In the following subsections, we briefly highlight the important contributions in security, privacy and trust domains of VANETs.

6.1.1 Security

The main contribution of Chapter 3 is to propose solution to support message credibility and availability by a voting-based consensus algorithm, PoQF. It serves the dual purpose of message validation and a competitive relay selection process based on probabilistic prediction of distance and channel quality between transmitter and receiver. It is proposed as a suitable solution to VANETs as it results in 11% and 15% higher security than PoS and PoET, respectively and is 68ms faster than PoET. More specifically, the proposed solution includes

- Voting based message validation mechanism
- Quality factor based multi-hop relay selection
- Scalable blockchain and consensus algorithm

Another relay selection algorithm, PoFL is presented in Chapter 4 which is based on blockchain-enabled FL. It results in 65.2% less time delay in message dissemination per hop and 8.2% higher message delivery ratio than PoQF. It uses machine learning enabled
smart contract to provide security in presence of both malicious and selfish nodes. The complete solution includes

- Smart contract based security against poisoning attack
- FL based multi-hop relay selection
- Blockchain approach for complementing FL

### 6.1.2 Privacy

One of the motivations in using FL for message dissemination is to ensure privacy. As concluded in Table 4.4, the proposed PoFL provides better privacy compared to the other relay selection approaches. To further propose solutions for privacy enhancement, PLS integrated with blockchain is studied in Chapter 5. It analyses a blockchain-enabled FD-NOMA based V2X system and evaluates its performance in presence of jamming and eavesdropping attack. The main contributions include

- Evaluating the lower bound of required data rate for FD-NOMA based vehicular network to support both PLS and blockchain in presence of jammers and eavesdroppers
- 8.2 Mbps increase in blockchain’s goodput with PLS as compared to without PLS

### 6.1.3 Trust

For establishing trust in the proposed solutions, incentive mechanisms are designed which motivate nodes to take honest actions as their best response strategy. In both Chapter 3 and Chapter 4, the incentive mechanisms distribute rewards to contributing nodes on the basis of virtual credit $CC$ paid by $ORG$. $CC$ is proposed to be inversely proportional to $Rep_{ORG}$ to form an integrated strategy of both price-based and reputation-based incentive scheme. The motivation behind $CC$ being paid by $ORG$ is to create a sustainable economic model for promoting safe driving conditions and healthy traffic flow. Ideally, the virtual credit earned and lost in a year will reflect a node's behaviour. A node which is involved in less number of incidents would have spent less credit in originating a message, leaving higher balance remaining in its credit wallet, which can be redeemed into annual road tax. The nodes will therefore be motivated to drive safely to avoid incidents and to earn credit by
positive cooperation. For supporting secure annual road tax calculation, the distribution of incentives is proposed to be recorded as immutable blocks in blockchain.

6.2 Future Work

The following subsections discuss potential future research areas which can extend the work presented in this thesis.

6.2.1 Energy Aware, UAV Assisted Blockchain and AI for IoV

This thesis focuses on message dissemination on road, such as traffic jam or incident. However, timely message dissemination is of crucial importance in case of disasters also. Vehicular infrastructures are often prone to destruction in case of disasters and alternate solutions are therefore required [169]. Recently, UAVs featured with small size, low weight, flexible mobility with aerial capabilities are recommended for mission critical applications such as rescue service in post-disaster situation. UAVs can also assist vehicular networks through Air-to-Air (A2A) and Air-to-Ground (A2G) communications when infrastructure is unavailable or connectivity is poor [169]. Therefore, integration of UAVs and IoV, as shown in Figure 6.1, is a promising solution for enhanced connectivity. Since connected UAVs also form MANETs, their security, trust and privacy challenges are similar to VANETs. Blockchain can be a potential approach to resolve these challenges and ensure decentralisation. Furthermore, AI techniques are integral part for an intelligent
and autonomous solution [170]. However, UAVs are low powered devices with limited storage. Both AI and blockchain solutions must be lightweight and compatible with energy and memory requirements of UAVs.

In future work, we plan to propose scalable solutions such as Tiny Machine Learning [171] for UAVs to reduce computational load and memory demands of existing on-device machine learning algorithms. UAVs can be utilised for on-device machine learning and to overcome the existing limitation of FL, i.e., variation in global model’s performance in heterogeneous networks. Also, we plan to continue our work on devising blockchain consensus with same security as PoW but less computation. A promising alternative to PoW is Riemann Zeta function [172]. We will analyse its suitability for both UAVs and IoV.

6.2.2 Quantum Resistant Cryptography

Blockchain adopts cryptography for producing hash functions and digital signatures to ensure privacy. Analysis of cryptographic schemes used with blockchain is out of the scope of this thesis. However, as a future work, the suitability of a cryptographic scheme with respect to the processing power of nodes and latency requirements of VANET can be studied. The progress in quantum computing has led to the possibility of attacks on cryptographic algorithms used with blockchain [173]. Also, as concluded in Chapter 5, co-existence of PLS and blockchain is not feasible to protect eavesdropping attack with low data rate or large number of eavesdroppers. Therefore, robust cryptographic schemes are required to preserve privacy. We plan to explore robust but computationally simple quantum-resistant cryptographic schemes, such as lattice-based schemes [174] and chameleon hashes [175], which can be used with blockchain and VANETs.

6.2.3 Network Layer Aspects of Blockchain

This thesis proposes implementation of blockchain on application layer. Additionally, some MAC layer and physical layer aspects are employed to analyse blockchain throughput in Chapter 3 and Chapter 5 respectively. However, blockchain is one of the potential solutions to enhance network layer security as well, which are not explored in this thesis. The network layer features of blockchain, such as Denial-of-Service resistance is
analysed in [176]. In [177], network layer is used in heterogenous IoT environment to select a consensus algorithm on the basis of network topology, number of nodes and communication throughput. Furthermore, an efficient technique to enhance network layer trust, i.e., Named Data Networking can be supported by blockchain for better security [178]. A prospect future work lies in investigating blockchain-based solutions to enhance network layer security in IoV.

6.2.4 Alternatives to Linear Distributed Ledger

The high probability of fork occurrence in blockchain of vehicular networks is discussed in Chapter 2. In Chapter 3 and Chapter 4, the hierarchical structure of blockchain consisting of horizontal keyblocks and parallel microblocks is proposed as a solution. However, synchronisation issues may still arise due to dynamicity in vehicular networks. Future research can explore other alternatives to linear ledger. One of the potential techniques is Directed Acyclic Graph (DAG) [75], in which a new block is not necessarily linked with its latest predecessor but could be connected to any of the previous blocks. Tangle and Hashgraph are two DAG based approaches under research as a substitute to traditional distributed ledger [179]. A suitable solution for multi-hop message dissemination among vehicle nodes can be made using Monoxide blockchain [180], which consists of zones of blocks instead of one single blockchain and relays transaction from one zone to another, similar to relaying a message from one hop to another.
Bibliography


A. Iqbal, A. S. Rajasekaran, G. S. Nikhil and M. Azees, “A Secure and Decentralized Blockchain Based EV Energy Trading Model Using Smart


Data,” in *Proc. of 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, April 2017.


Appendix A

Simulation Tools

Table A.1 lists the simulation tools used to validate results of the proposed solutions described in Chapter 3, Chapter 4 and Chapter 5. Each simulation tool is described below.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Simulation Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 3</td>
<td>OMNeT++, VeINS, SUMO</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>OMNeT++, VeINS, SUMO, Tensorflow (Python)</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>OMNeT++, VeINS, SUMO, MATLAB</td>
</tr>
</tbody>
</table>

Table A.1: Simulation tools used for each of the proposed solution.

A.1 OMNeT++

OMNeT++ is a modular, C++ based framework primarily used for simulating communication networks [181]. It includes an integrated development and graphical runtime environment. It is an open-source tool available at https://www.omnetpp.org. The specific simulation parameters used to run Veins for each of the proposed solutions are listed in Simulated Performance Analysis section of each Chapter. The general parameters used in every solution are shown in Table A.2. The SHA-256 algorithm for generating cryptographic blocks in OMNeT++ is supported by Crypto++ library [182].

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>IEEE 802.11p</td>
<td>Encryption</td>
<td>SHA-256</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Krauss</td>
<td>$L$</td>
<td>756 bytes</td>
</tr>
<tr>
<td>Beacon frequency</td>
<td>0.1 s</td>
<td>$n_{hop}^{max}$</td>
<td>6</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>3</td>
<td>$R$</td>
<td>250 m</td>
</tr>
</tbody>
</table>

Table A.2: Simulation Parameters.

A.2 Simulation of Urban Mobility (SUMO)

SUMO is a road traffic simulator which offers a socket-based interface to interact with external applications such as OMNeT++ [183]. It can be accessed as open-source at
https://www.eclipse.org/sumo/. It allows to load maps, design routes, control speed limits and distances among nodes. Figure A.1 shows the simulation map of University of Sussex campus obtained from https://www.openstreetmap.org and used in experiments of the proposed solution.

Figure A.1: Simulation map of University of Sussex campus.

A.3 Vehicles in Network Simulation (VeINS)

VeINS is also an open-source framework for running vehicular network simulations in OMNeT++ integrated with SUMO [184]. The original VeINS setup simulates an accident upon which the oncoming nodes receive a message. It can be accessed at https://veins.car2x.org. Figure A.2 shows the Graphical User Interface (GUI) of VeINS running in OMNeT++.
A.4 Tensorflow (Python)

Tensorflow is an open-source library offering training and inference of deep neural networks [185]. It is available at www.tensorflow.org. It supports multiple languages including C++ and Python. The proposed solution of FL described in Chapter 4 is implemented using Tensorflow in Python. For PoFL based message dissemination, the execution of global model is simulated by calling Python function in OMNeT++, supported by Python.h library [186].

A.5 MATLAB

MATLAB is a software package which allows to develop algorithms and models mostly used for mathematical computations and iterative analysis. It is commonly used by researchers for PLS analysis [187] - [188]. Many special mathematical computations, such as gamma function \( \Gamma(.) \) and exponential integral \( E_1(.) \) are available as built-in functions in MATLAB.
Appendix B

Code of blockchain implementation in OMNET++

//Define_Module(Block);

Block::Block(dataStruct transactionData){
    Amount = transactionData.Amount;
    Comment = transactionData.Comment;
    transactionIndex = 0;
    Payer = transactionData.Payer;
    Recver = transactionData.Recver;
    sPrevHash = "None";
    sCurrentHash = _CalculateHash();
};

std::string Block::GetHash(){
    return sPrevHash;
};

inline std::string Block::_CalculateHash() const {
    std::stringstream ss;
    ss << Amount << Comment << transactionIndex << Payer << Recver << sPrevHash;

    return sha256(ss.str());
};
Appendix C

Code Snippet of OMNeT++ integrating Python program

CPyObject pModule=PyImport_Import(pName);
PyObject *arg;
PyObject *result;
    if (pModule)
    {
        CPyObject pFunc= PyObject_GetAttrString(pModule, "getModel");
        if(pFunc && PyCallable_Check(pFunc))
        {
            arg=Py_BuildValue("(i,i,d,d,d)",hop+1,d_is,speed_data,dir,gammai);
            result=PyObject_CallObject(pFunc, arg);
            score=PyFloat_AsDouble(result);
        }
    }
Appendix D

MATLAB Code for FD-NOMA

clc;
clear all;
for it=1:100000

%======================================================================
%           Full Duplex Non Orthogonal Multiple Access (NOMA) Simulation
%       Version 1: Encode only: Transmitter side,
%======================================================================
%       Contact: Ferheen Ayaz (f.ayaz@sussex.ac.uk)
%======================================================================
% n0.0f bits for transmit signal: 4 bits only - REMOVE '/25' for 100 bits
% as original SMT/TNY codes;
  TxBits_n = 100/25;
% distance from Sender i to Receiver j1, Receiver j2,Receiver j3,
% Receiverj4
% Assuming maximum distance as 1000, minimum distance as 4, for
attenuation calculation purposes
  Dstjj1 = (1000-4).*rand + 4;
  Dstjj2 = (1000-4).*rand + 4;
  Dstjj3 = (1000-4).*rand + 4;
  Dstjj4 = (1000-4).*rand + 4;
  MaxDsttoveh = 1000;
  Dst=[Dstji Dstj1 Dstj2 Dstj3 Dstj4];
  Dst=sort(Dst);
  %Arranging for SIC
  Dstj1=Dst(4);
  Dstj2=Dst(3);
  Dstj3=Dst(2);
  Dstj4=Dst(1);

%Tx Power P=Pi=Pl
  TotPwrS_00 = 0.100;
  Pwrj1 = 0.100;
Pwrj2 = 0.100;
Pwrj3 = 0.100;
Pwrj4=0.100;
eta=0.1; %Self-interference co-efficient
alpha=-3; %Path loss

%%%Create random binary messages/signals from Sender

Sgni = rand(1,TxBits_n) > 0.5;
Sgnj1 = rand(1,TxBits_n) > 0.5;
Sgnj2 = rand(1,TxBits_n) > 0.5;
Sgnj3 = rand(1,TxBits_n) > 0.5;
Sgnj4 = rand(1,TxBits_n) > 0.5;

%%%Superposition Encoding

Enc_Xi = sqrt(TotPwrS_00)*Sgni;
Enc_Xj1 = sqrt(Pwrj1)*Sgnj1;
Enc_Xj2 = sqrt(Pwrj2)*Sgnj2;
Enc_Xj3 = sqrt(Pwrj3)*Sgnj3;
Enc_Xj4 = sqrt(Pwrj4)*Sgnj4;

%%%Received signals for all vehicles

% Adding Gaussian Noise: use 'randn' instead of 'rand':
'randn' generates numbers in \([-\infty, +\infty]\), normally distributed (Gaussian);
% Mean is zero, but with strong concentration in \([-1 \text{ to } +1]\);
% 'rand' generates numbers in \([0 \text{ to } 1]\), uniformly distributed;
% Mean is 0.5
% Noise

\[
\begin{align*}
\text{Noise}j1 &= \frac{\text{randn}(1, \text{TxBits}_n)}{\text{NoiseReduc}_0}; \\
\text{Noise}j2 &= \frac{\text{randn}(1, \text{TxBits}_n)}{\text{NoiseReduc}_0}; \\
\text{Noise}j3 &= \frac{\text{randn}(1, \text{TxBits}_n)}{\text{NoiseReduc}_0}; \\
\text{Noise}j4 &= \frac{\text{randn}(1, \text{TxBits}_n)}{\text{NoiseReduc}_0}; \\
\end{align*}
\]

% Channel Coefficient
\[
\begin{align*}
hj1 &= \sqrt{\left(Dstj1 \cdot ^{-1} \cdot \alpha \right) \cdot \left(\text{randn}(1, \text{TxBits}_n) + \text{i} \cdot \text{randn}(1, \text{TxBits}_n)\right)}; \\
hj2 &= \sqrt{\left(Dstj2 \cdot ^{-1} \cdot \alpha \right) \cdot \left(\text{randn}(1, \text{TxBits}_n) + \text{i} \cdot \text{randn}(1, \text{TxBits}_n)\right)}; \\
hj3 &= \sqrt{\left(Dstj3 \cdot ^{-1} \cdot \alpha \right) \cdot \left(\text{randn}(1, \text{TxBits}_n) + \text{i} \cdot \text{randn}(1, \text{TxBits}_n)\right)}; \\
hj4 &= \sqrt{\left(Dstj4 \cdot ^{-1} \cdot \alpha \right) \cdot \left(\text{randn}(1, \text{TxBits}_n) + \text{i} \cdot \text{randn}(1, \text{TxBits}_n)\right)}; \\
\end{align*}
\]

%%Signal received by each vehicle

\[
\begin{align*}
\text{RxSgn}j5 &= \text{Enc}_X1 \cdot hji + \text{Noise}j1 + (\text{Enc}_X1 \cdot hj1) + (\text{Enc}_X2 \cdot hj2) + (\text{Enc}_X3 \cdot hj3) + (\text{Enc}_X4 \cdot hj4) + (\text{Enc}_X5 \cdot hj5) + \text{SI}; \\
\text{RxSgn}j4 &= \text{Enc}_X1 \cdot hji + \text{Noise} + (\text{Enc}_X1 \cdot hj1) + (\text{Enc}_X2 \cdot hj2) + (\text{Enc}_X3 \cdot hj3) + (\text{Enc}_X4 \cdot hj4) + \text{SI}; \\
\text{RxSgn}j3 &= \text{Enc}_X1 \cdot hji + \text{Noise}j2 + (\text{Enc}_X1 \cdot hj1) + (\text{Enc}_X2 \cdot hj2) + (\text{Enc}_X3 \cdot hj3) + \text{SI}; \\
\text{RxSgn}j2 &= \text{Enc}_X1 \cdot hji + \text{Noise}j3 + (\text{Enc}_X2 \cdot hj2) + (\text{Enc}_X1 \cdot hj1) + \text{SI}; \\
\text{RxSgn}j1 &= \text{Enc}_X1 \cdot hji + \text{Noise}j4 + (\text{Enc}_X1 \cdot hj1) + \text{SI}; \\
\end{align*}
\]