Blockchain-Enabled Data Sharing in Connected Autonomous Vehicles for Heterogeneous Networks

Ali Hussain Khan[1], Naveed UL Hassan[1(✉)], Chuadhry Mujeeb Ahmed[2], Zartash Afzal Uzmi[1], and Chau Yuen[3]

[1] Lahore University of Management Sciences (LUMS), Lahore, Pakistan
{ali_k,naveed.hassan,zartash}@lums.edu.pk
[2] University of Newcastle, Newcastle, UK
mujeeb.ahmed@newcastle.ac.uk
[3] Nanyang Technological University (NTU), Singapore, Singapore
chau.yuen@ntu.edu.sg

Abstract. In this paper, we consider the use of blockchain for a challenging Connected Autonomous Vehicles (CAV) application scenario of data sharing that has stringent security requirements. We discuss enhanced delegated Proof-of-Stake (dPoS) consensus and combine it with three different reputation schemes, which are Multi-Weight-Subjective-Logic (MWSL), beta, and sigmoid-based reputation schemes. To determine malicious miners in the system, we first compute the overall latency of block generation as the sum of communication, computational, information propagation and queuing latency. We then evaluate the performance of this scheme under simple and orchestrated adversary attack models. We do the same analysis for a heterogeneous network, where we validate our results with a 5G/6G hybrid network. Our results indicate that in large scale networks, MWSL reputation based enhanced dPoS scheme can detect orchestrated attacks in few seconds for 6G networks. Heterogeneous deployment schemes also perform relatively well. In comparison, 4G and 5G perform poorly, and might not be suitable for blockchain implementations.

Keywords: Blockchain · Simple and orchestrated adversary · Heterogeneous network

1 Introduction

Completely autonomous vehicles (AVs) are embedded with multiple sensors and electronic control units (ECUs) [9] which communicate with each other to facilitate the intra-vehicle decisions. AVs communicate with other AVs as well as the infrastructure to share their data for streamlined traffic flow. AVs also communicate with remote cloud and servers for storage and computation requirements. Given the complexity of the application, the large amount of data being shared and the millions of lines of code, the attack surface is huge [10]. There are privacy,

J. Zhou et al. (Eds.): ACNS 2023 Workshops, LNCS 13907, pp. 215–230, 2023.
https://doi.org/10.1007/978-3-031-41181-6_12

data integrity, auditability and non-repudiation concerns originating from compromised data. Recent proposals about Connected AVs (CAVs) communication frameworks depend on centralized architecture [4] which have single point-of-failure along with privacy and security concerns.

Recently, blockchain has emerged as an all-in-one solution to all the security and privacy related needs of many applications [2]. In a blockchain, cryptography and hash functions are used together to form a chain of blocks where each block is added to the chain after achieving consensus within the network in a decentralized way. These consensus mechanisms ensure the trustless nature of blockchain by making all the entities verify the state of the network themselves. Popular consensus mechanisms in blockchain are Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (dPoS) and Practical Byzantine Fault Tolerance (PBFT) [11].

Previously, blockchain has been utilized in data sharing among AVs. In [12], the authors presented vehicles as blockchain nodes which brought limitations. In [7], the authors presented a framework where road side units (RSUs) were blockchain nodes and vehicles utilized RSUs for mobile edge computing and the high convergence latency of public network is mitigated by using a consortium network. The recurring limitation in these blockchain-based data-sharing schemes is that although all of them highlighted the long convergence times, none of them quantified the latency of their scheme.

The data shared between AVs and network has to pass through multiple RSUs which may be operating on different cellular technologies with backward and forward compatibility. Currently, the commercial deployment of 5G is observed to be in phases. Therefore, 5G is expected to coexist with 4G for a long period of time. Recently, many works have studied the coexistence and interworking of 4G and 5G network. In [14], the authors claim that the network shift will be from 4G to 5G Non-Standalone (NSA), and then to 5G Standalone (SA), where, first the Radio Access Network (RAN) infrastructure will move from 4G to 5G followed by the control infrastructure. In [15], the authors state that 4G and 5G will coexist initially. This is because 4G has more coverage than 5G. In that case, specific applications that prefer coverage over data rates and latency can use 4G in the same frequency band as 5G. These works also suggest that there is backward compatibility between 4G and 5G i.e., depending on available resources, the same infrastructure may provide 4G or 5G connectivity.

The security measures vary across networks, which may pose significant challenges in data sharing among CAV applications [8]. Blockchain can be added as an over-the-top solution on wireless networks. Because of the recent large-scale move towards wireless networks, research community has started to focus its interest towards blockchain in wireless networks. Blockchain provides an elegant solution to these data sharing challenges. However, the difference in the network speeds might still impact the overall security of the application. Therefore, in this paper, we perform an end-to-end latency analysis of an enhanced dPoS scheme [6] with the objective of analyzing the security of CAV applications. This dPoS scheme is selected because it has several safeguards to detect malicious activities

and it can also be deployed on several types of networks. It is also not computationally intensive like PoW and PoS, and therefore, convergence latency can be tamed by the choice of network.

In this work, we consider a CAV application scenario for studying and providing a solution to its security requirements. The employment of blockchain in CAV is discussed in great detail in [9]. The application is highly dynamic and some decisions require extremely strict deadlines. We consider a data sharing scenario where the data shared between the AVs and RSUs is stored in a blockchain, which employs reputation-based enhanced dPoS consensus scheme [6]. The major contributions of this work are as follows:

1. We study the data sharing among CAVs in a heterogeneous network with different coexisting networks. This is a particularly important problem, because the shift from one generation to another is incremental (as discussed before for the case of 5G deployment), which has not been studied in the literature.
2. We present an end-to-end latency of the considered enhanced dPoS scheme. This is done by presenting analytical expressions and performing numerical simulations.
3. Reputation management is a robust way to characterize an entity as malicious or honest based on their behavior. Therefore, it is a vital component of blockchain consensus. For CAV application, it is especially important because data integrity is a critical requirement. We present a security analysis of this scheme where we evaluate three different reputation models i.e., Multi-Weight Subjective Logic (MWSL) [6], beta [5], and sigmoid-based [1] reputation models, and show their performance in terms of aggressiveness towards malicious behavior in a simple and orchestrated adversary model.
4. Based on aggresiveness towards malicious miner detection, we derive the time required to detect and ultimately remove malicious miners in both homogeneous and heterogeneous networks for these reputation schemes.

The rest of the paper is organized as follows. In Sect. 2, we discuss the blockchain-enabled data sharing scenario. In Sect. 3, we present an end-to-end latency analysis for both homogeneous and heterogeneous networks. In Sect. 4, we discuss our simulation setup and present the results based on that. Finally, in Sect. 5, we conclude the paper.

2 Blockchain Based CAV Application Scenario

In this section, we describe a blockchain based CAV application scenario.

2.1 System Model

The data sharing system consists of a Trusted Authority (TA), RSUs and CAVs.

TA: The TA is responsible for registering the CAVs and RSUs in the system. CAVs and RSUs submit their registration information and get their public/private key pairs and the digital certificates. The TA also registers RSUs as miner candidates based on their reputation values stored on the blockchain.

RSUs: The infrastructure elements in the system are RSUs, which are deployed on the sides of the roads to assist in data sharing and storing. RSUs have high computational and storage resources that enable them to act as blockchain nodes. They are responsible for carrying out consensus in the system and updating the blockchain after every cycle along with storing the blockchain and the reputation matrix. We consider a heterogeneous network, where some RSUs are connected with 5G while others are connected with 6G.

CAVs: CAVs share data with each other and share the data sharing as a transaction to the nearest RSU. After the consensus, when the block is added to the blockchain, the CAVs download the block. Based on the correctness of the uploaded data, they update the reputation values of the RSUs.

2.2 Trust Model

Trust is a very important parameter in a blockchain network which entails the credibility of an entity in the network. In the CAV application, the network is dynamic and it is necessary to choose a robust trust model. The trust model used in this work is based on the reputations of the consensus nodes. Reputation depends on the extent of positive and negative interactions of vehicles with different RSUs. Positive interactions increase the reputation and negative interactions decrease the reputation. Positive and negative interactions are compiled using reputation models. Higher reputation translates into more trustworthiness and vice versa. Reputation plays an integral role when it comes to miner voting as it is an indicator that can be used by the stakeholders to prefer specific miners over the others.

2.3 Blockchain-Based Data Sharing CAV Scenario

Here we describe the blockchain-based data sharing scenario based on the enhanced dPoS [6] as summarized in Fig. 1.

Step 1: The CAVs having high stakes in the network act as stakeholders and RSUs as delegates. Stakeholders vote for their choice of miners. Out of the elected miners, a predefined number is selected as active miners, which act as block managers for the next few time instants, and the rest as standby miners. Block managers are the same as leaders in PBFT and its derivatives. Standby miners improve the security of the system by increasing the number of miners that verify the data. Verifiers are divided into different types based on their reputations. The CAVs share data with each other via Vehicle-to-Vehicle (V2V) communication. This data sharing record is sent to the nearest RSU via Vehicle-to-Infrastructure (V2I) communication. In a heterogeneous network, RSUs may be connected to different networks. Then this data is routed to the block manager.

Step 2: The block manager forms an unverified block out of this data and encodes smart contracts based on the number of verifier types in the system. Then this block is broadcasted to the whole network of verifiers. The verifiers

work on the smart contracts based on their type, as doing so maximizes their utility. After verification, the results are sent to their local neighborhood, where these results are audited by verifiers in their one hop vicinity. After verification, this block is sent back to the block manager.

Step 3: Block manager receives the verified block from all the miners and verify whether 2/3 of the miners agree on the block creation or not. If the consensus is achieved, the block manager broadcasts this block to all the RSUs in the system to add to their local blockchains.

Step 4: After the block is added to the blockchain, AVs download the latest data block and check whether their transaction is added correctly. Based on that, they calculate the reputation of the respective miner.

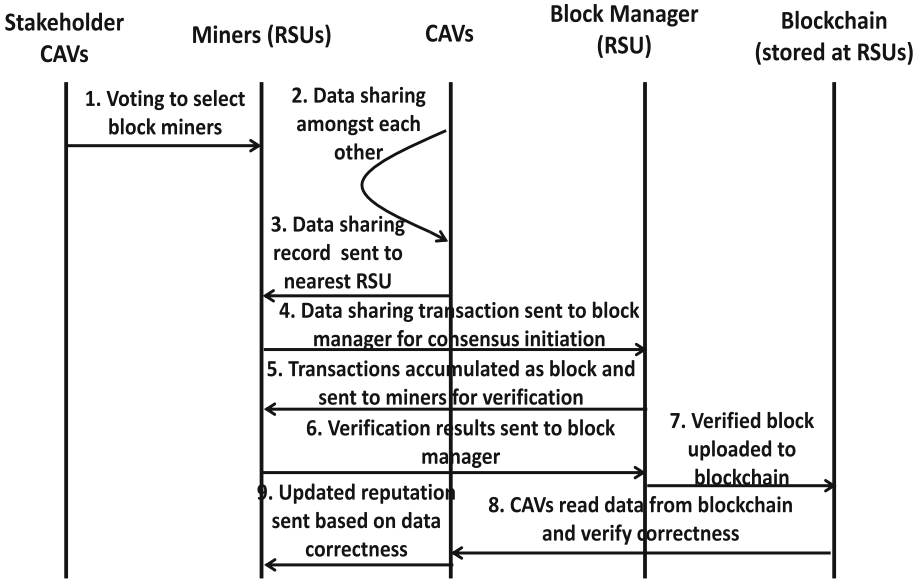

Fig. 1. Information flow of enhanced dPoS algorithm for data sharing

In the next section, we provide a latency and security analysis of this blockchain scheme for homogeneous and heterogeneous network conditions. It is important to understand how quickly malicious activities can be detected in CAV applications and this analysis provides a basis for that.

3 Latency and Security Analysis for CAV Data Sharing

In this section, we present latency and security analysis of blockchain-based CAV data sharing scenario. We present the analysis for both homogeneous and

Table 1. Description of Relevant Notations.

Notation	Description
M	Total active and standby miners in the system
k	Number of active miners in the system
N_{veh}	Number of AVs in the system
I_{vote}	Vote Size
r_v	Download and upload data rates of vehicles
r_m	Download and upload data rates of miners
r_l	Download and upload data rates of miners connected to lower generation network in heterogeneous network
r_h	Download and upload data rates of miners connected to higher generation network in heterogeneous network
r_{avg}	Average download and upload data rates of miners in heterogeneous network
P_{hl}	Probability of having a lower genration node at the outbound link of a higher generation node
I_k^d	Data block size before verification
I_k^r	Reputation block size
I_k^{SC}	Smart contract size
β	Number of types of verifiers in the system
N_{hops}	Maximum end-to-end number of hops
α	Averaging factor for the number of hops
N_{vd}	Number of RSUs with vehicular data
$\frac{Task_m^k}{c_m^k}$	Computational requirement of the smart contract relative to computations per second of a miner
O_k	Block size after verification
I_k^{ver}	Block verification overhead
$\frac{C_{inst}}{c_m^k}$	Computational requirement of the forwarding flow relative to computations per second of a miner

heterogeneous networks. As mentioned before, practical networks will have heterogeneous network nodes. Therefore, an end-to-end latency analysis in such networks is an important problem which has not been discussed previously in the literature. The description of different notations used in this paper are given in Table 1.

In this section, we compute the overall latency of block generation i.e., one round of consensus and block update for both homogeneous and heterogeneous networks, which is the sum of transmission, computational, and information diffusion latency. The computational latency comes from the computations required for parsing the IP headers and relaying the block. The information diffusion latency comes from broadcasting the block in the complete network. For heterogeneous networks, we also consider queuing latency as when we move from higher generation to lower generation nodes, packets are buffered at the lower

generation node due to differences in data rates, and there is queuing latency. The data sharing scenario as described above is further split into sub-steps and the latency for each sub-step is analyzed as follows.

3.1 Latency Analysis for Homogeneous Networks

Step 1a: CAVs participate in a decentralized voting process to determine the miners. Voting results are broadcast. There is vote broadcast and vote tallying in this sub-step. The transmission latency (averaged over k rounds) for that is

$$\frac{M \times I_{vote}}{r_v \times k}$$

The computation latency (averaged over k rounds) is

$$\frac{C_{inst}}{c_m^k} \times N_{hops} + \frac{M^2}{k \times c_m^k}$$

and the diffusion latency (averaged over k rounds) is

$$\frac{(M \times I_{vote}) \times N_{hops}}{r_m \times k}$$

Step 1b: CAVs exchange data and share transactions (data and latest reputation scores) with the nearest RSU. The transmission latency of this process is

$$\frac{\frac{I_k^d + I_k^r}{N_{veh}}}{r_v}$$

The computation latency associated with this substep is

$$\frac{C_{inst}}{c_m^k}$$

The information propagation latency associated with this sub-step is zero because there is no broadcast of information.

Step 1c: RSUs route these transactions to the block manager of current round. There is no transmission latency. The computation latency is given as

$$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$$

and the information propagation latency is given as

$$\frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_m}$$

Step 2a: Block manager forms an unverified block and encodes smart contracts based on number of verifier groups. The unverified block and the smart contracts are broadcast to the miners. The transmission latency of this process is

$$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_m}$$

The computation latency is

$$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$$

and the information propagation latency is

$$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_m} \times N_{hops} \times \alpha$$

Step 2b: Miners work on the smart contract and get the block verified in their local neighborhood. There is no transmission latency of this process. The computational latency is given as

$$\frac{Task_m^k}{c_m^k}$$

and the information propagation latency associated with relaying the verification result to one hop neighborhood is

$$\frac{I_k^{ver}}{r_m}$$

Step 2c: Verified block is sent back to the block manager. There is no transmission latency associated with this sub-step. The computation latency is given as

$$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$$

and the information propagation latency is given as

$$\frac{(I_k^d + I_k^r + I_k^{ver} + I_k^{SC})}{r_m} \times N_{hops} \times \alpha$$

Step 3: Block manager broadcasts the verified block to the network to be added to the local blockchain copies. The transmission latency of this process is

$$\frac{O_k}{r_m}$$

The computation latency of this process is

$$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$$

and the information propagation latency is

$$\frac{O_k}{r_m} \times N_{hops} \times \alpha$$

Step 4a: Finally, CAVs download the latest data block and reputation opinions of the RSUs. This incurs only a transmission latency of

$$\frac{O_k}{r_v}$$

and no computation or information propagation latency.

Step 4b: Based on the correctness of the uploaded data, new reputation values are calculated for the relevant RSUs, which requires minimal computational latency and zero transmission, and information diffusion latency.

3.2 Latency Analysis for Heterogeneous Network

For heterogeneous network, the computational latency is the same as that in homogeneous networks. The transmission and propagation latency change based on the average data rate based on the number of nodes from different networks. The queuing latency comes from buffering of packets when we a higher generation RSU has a lower generation RSU on its outbound. The latencies for heterogeneous network are as follows.

Step 1a: The transmission latency of this sub-step is the same as that for homogeneous networks and the diffusion latency is

$$\frac{(M \times I_{vote}) \times N_{hops}}{r_{avg} \times k}$$

The queuing latency of this sub-step is

$$P_{hl} \times \frac{(M \times I_{vote}) \times N_{hops}}{r_h \times k} \times (\frac{r_h}{r_l} - 1)$$

Step 1b: The transmission latency of this process is the same as that for homogeneous networks. The information propagation latency and the queuing latency associated with this sub-step is zero because there is no broadcast of information.

Step 1c: There is no transmission latency. The information propagation latency is given as

$$\frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_{avg}}$$

and the queuing latency is given as

$$P_{hl} \times \frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_h} \times (\frac{r_h}{r_l} - 1)$$

Step 2a: The transmission latency of this process is

$$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_{avg}}$$

The information propagation latency is

$$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_m} \times N_{hops} \times \alpha$$

and the queuing latency is

$$P_{hl} \times \frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_h} \times N_{hops} \times \alpha \times (\frac{r_h}{r_l} - 1)$$

Step 2b: There are no transmission and queuing latencies for this process. The information latency for this process is

$$\frac{I_k^{ver}}{ravg}$$

Step 2c: There is no transmission latency of this process and the information propagation latency is given as

$$\frac{(I_k^d + I_k^r + I_k^{ver} + I_k^{SC})}{r_{avg}} \times N_{hops} \times \alpha$$

and the queuing latency is given as

$$P_{hl} \times \frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_h} \times (\frac{r_h}{r_l} - 1)$$

Step 3: The transmission latency of this process is

$$\frac{O_k}{r_{avg}}$$

and the information propagation latency is

$$\frac{O_k}{r_{avg}} \times N_{hops} \times \alpha$$

The queuing latency of this process is

$$P_{hl} \times \frac{O_k}{r_h} \times N_{hops} \times \alpha \times (\frac{r_h}{r_l} - 1)$$

Step 4: The latency of this step is the same for both homogeneous and heterogeneous networks as there is no role of RSUs in this step. The queuing latency is also zero (Table 2).

Table 2. Latency of One Complete Cycle

Steps	Description	Transmission Latency	Computation Latency	Information Propagation Latency	Queuing Latency (for heterogeneous networks)
Step 1	CAVs participate in a voting process to determine the miners. Voting results are broadcast	$\frac{M \times I_{vote}}{r_v \times k}$	$\frac{C_{inst}}{c_m^k} \times N_{hops} + \frac{M^2}{k \times c_m^k}$	$\frac{(M \times I_{vote}) \times N_{hops}}{r_m(r_{avg\ \ for\ \ het}) \times k}$	$L_{21} \times \frac{(M \times I_{vote}) \times N_{hops}}{r_2 \times k} \times (\frac{r_2}{r_1} - 1)$
	CAVs exchange data and share transactions (data and latest reputation scores) with the nearest RSU	$\frac{\frac{I_k^d + I_k^r}{N_{veh}}}{r_v}$	$\frac{C_{inst}}{c_m^k}$	0	0
	RSUs route transactions to the block manager of current round	0	$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$	$\frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_m(r_{avg\ \ for\ \ het})}$	$L_{21} \times \frac{\frac{(I_k^d + I_k^r)}{N_{vd}} \times N_{hops} \times \alpha}{r_2} \times (\frac{r_2}{r_1} - 1)$
Step 2	The unverified block and the smart contracts (SC) are broadcast to the miners	$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_m}$	$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$	$\frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_m(r_{avg\ \ for\ \ het})} \times N_{hops} \times \alpha$	$L_{21} \times \frac{I_k^d + I_k^r + \beta I_k^{SC}}{r_2} \times N_{hops} \times \alpha \times (\frac{r_2}{r_1} - 1)$
	Miners work on the SC and get the block verified in their local neighborhood	0	$\frac{Task_m^k}{c_m^k}$	$\frac{I_k^{ver}}{r_m}$	0
	Verified block is sent back to the block manager	0	$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$	$\frac{(I_k^d + I_k^r + I_k^{ver} + I_k^{SC})}{r_m(r_{avg\ \ for\ \ het})} \times N_{hops} \times \alpha$	$L_{21} \times \frac{(I_k^d + I_k^r + I_k^{ver} + I_k^{SC})}{r_2} \times N_{hops} \times \alpha \times (\frac{r_2}{r_1} - 1)$
Step 3	Block manager broadcasts the verified block to the network to be added to the local blockchain copies	$\frac{O_k}{r_m}$	$\frac{C_{inst}}{c_m^k} \times N_{hops} \times \alpha$	$\frac{O_k}{r_m(r_{avg\ \ for\ \ het})} \times N_{hops} \times \alpha$	$L_{21} \times \frac{O_k}{r_2} \times N_{hops} \times \alpha \times (\frac{r_2}{r_1} - 1)$
Step 4	CAVs download the latest data block and reputation opinions	$\frac{O_k}{r_v}$	0	0	0
	CAVs calculate reputation updates	0	simple arithmetic computations	0	0

3.3 Threat Model and Security Analysis

In the described blockchain model, there are multiple threat dimensions. We are assuming that the only trustworthy party is the TA while the RSUs and CAVs can be compromised.

Malicious RSUs: Malicious RSUs can add incorrect verifications of the data to sabotage the data to be added to the blockchain. Malicious RSUs can also collude with the malicious vehicles to maintain a high reputation and have higher chances to stay in the system.

Malicious CAVs: Malicious high stake stakeholder CAVs can vote for their choice of malicious RSUs to serve as block managers or verifiers in the upcoming block verifications. CAVs can also collude with malicious RSUs to make them stay in the system longer by giving positive reputation opinions.

Based on the above threat model, we analyze the security of the system. Legitimate transactions, when reported incorrectly will be audited by the CAVs when they are added to the blockchain. They can report them as an incorrect transaction which will decrease the miner reputation. Since the data represents sensor readings which are very important for driving decisions, the data is very sensitive. Whenever data is uploaded to blockchain, the credibility of data is verified. Based on that, invalid data is identified and ignored.

The reputation value varies from 0 to 1 and the reputation threshold for honest and malicious miners is set to be 0.5. Different reputation models have different update sensitivities. In this analysis, we show that a malicious RSU will be eventually eliminated by the system as its reputation falls below the threshold. We consider three different reputation models i.e., MWSL reputation model [6], beta reputation model [5], and sigmoid-based reputation model [1] with the blockchain based dPoS scheme, to observe the number of rounds in which malicious activities can be detected which impacts system security. The upload and download speeds are different in 4G, 5G, and 6G networks, which impacts the transmission, information propagation and queuing latency of various steps. Therefore, through this analysis we can easily determine the overall time required to generate a single block in different networks. Based on the above analysis, we compare the time required to detect malicious miners in the system for all three networks. This will impact the network and security performance of the system.

4 Numerical Case Study

In this section, we design a numerical case study to apply our framework to determine end-to-end latency and security of blockchain-based CAV application in 4G, 5G, and 6G networks as well as a heterogeneous 5G/6G network.

4.1 Simulation Setup

We consider a large-scale CAV system with 10000 AVs and 10000 RSUs uniformly distributed in a 150 km^2 area. The download and upload data rates of

CAVs as well as RSUs are 10 Mbps for 4G, 500 Mbps for 5G and 100 Gbps for 6G. We consider an unverified block size, vote size, reputation block size, and the smart contract size to be 5MB, 100KB, 150KB and 150KB respectively. We consider 10 types of verifiers. There are 199 active miners and the number of RSUs with vehicular data in each round is a uniformly distributed random variable between [1000, 4000]. Task computational latency is assumed to be a constant value of 0.5 s in each case. The computational latency of each forwarding flow is assumed to be 10 µs [13]. The values of α and β are respectively set as 0.75 and 10. The maximum end-to-end number of hops are calculated assuming a coverage range of 250 m using [3], which comes out to be 97.

In heterogeneous networks, we assume that the nodes have different data rates. We assume that there are 50% 6G RSUs with overall 10% network nodes where there is a 5G node present at the outbound of 6G node. The rest of the nodes are 5G nodes. We call this case as Het1 in simulations. We also consider 25% 6G RSUs and because the probability of 5G node at the outbound of 6G node increases as overall 6G nodes in the system decrease, we consider that 15% 6G nodes have 5G nodes at their outbound. We call this case Het2. The remaining nodes are 5G nodes. Using these parameters, we calculate the time required for detecting malicious miners under different reputation models in different networks under simple and orchestrated attacks with different percentages of colluding AVs ranging between 0% to 50%.

In the simple attack, a malicious miner remains honest for first 20 rounds and then switches to bad behavior. In the orchestrated adversary model, the miner also acts honestly for the first 20 interactions to gain reputation. However, it then behaves maliciously and honestly for 15 and 5 interactions interchangeably. We determine the number of cycles required to detect malicious miners under different reputation schemes and multiply it with one cycle latency to calculate the minimum latency required to detect malicious miners.

4.2 Simulation Results

In Fig. 2, we compare the total time required for malicious miner detection in 4G, 5G, 6G, Het1 and Het2 networks for MWSL (M1), beta (M2) and sigmoid (M3) reputation models at different collusion rates (between malicious RSUs and CAVs) for simple adversary model. We simulated the three reputation models for both adversary models and calculated the number of interactions it will take to fall under the reputation threshold. In Fig. 3, we repeated the same results for the orchestrated adversary model. In both figures, due to the extremely large time needed by 4G network, we plot the y-axis on a logarithmic scale. This is a stacked bar graph, where the total height of the bar indicates the total latency in 4G network, the sum of first four, three and two segments indicate the latency in 5G network, Het2 network and Het1 network respectively, while the bottom portion indicates the latency in 6G network.

6G network can detect malicious miners in few seconds, as compared to 4G and 5G. It is clear that 4G and 5G are not feasible for such a system. However, the Het1 and Het2 networks show significant improvement over 5G performance. If we compare the reputation models, we observe that for lower collusion rates, in a

simple attack scenario, all the reputation models combined with dPoS blockchain consensus have similar detection performance. For large collusion rates, there is a significant difference between the detection latency and M1 performs much better as compared to M2 and M3. At 40% collusion, M1 detects malicious miner in 48.7 s in 6G system. In Het1 and Het2, M1 detects malicious miners in 150.8 s and 200.3 s respectively. M2 and M3 reputation models combined with dPoS blockchain take 78.58 s and 79.23 s respectively for 6G. Het1 and Het2 perform the detection in 243.25 s & 323.1 s and 245.26 s & 325.78 s respectively. For 50% collusion rate, M1 takes 69.49 s, 215.1 s and 285.72 s for 6G, Het1 and Het2 respectively. M2 converges after an extremely large number of interactions, while M3 fails to detect the malicious miner. The results for 50% collusion are not plotted on the graph.

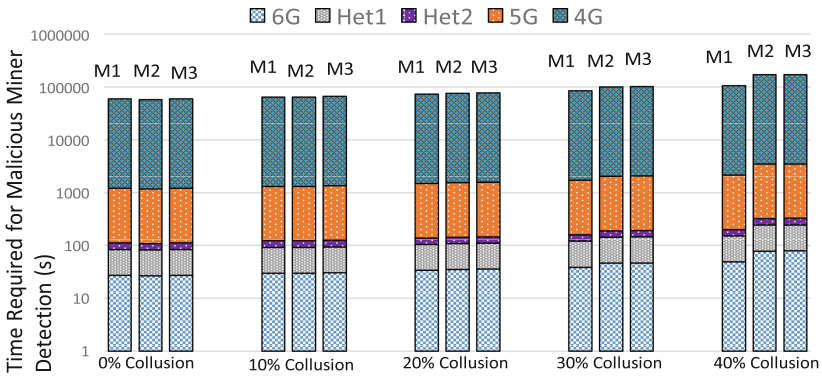

Fig. 2. Simple Adversary Model: Time required to detect malicious miner

For the orchestrated adversary attack shown in Fig. 3, at lower collusion rates, the three models show very similar behavior. However, even at 20% collusion rates, the time required for M1 becomes 43.5 s, 134.69 s and 178.91 s for 6G, Het1 and Het2 respectively. On the other hand, for M2 and M3, detection time is much larger at 61 s, 188.97 s & 251 s and 61.7 s, 190.98 s & 253.68 s respectively. At 30% collusion, M1 detects a malicious miner in 56.5 s, 174.9 s & 232.6 s whereas M2 detects it at more than triple the time i.e., 201.3 s, 623.2 s & 827.8 s. M3 has much higher latency than the upper limit of 400 interactions, which was set as a failure limit (the limit assumed in the simulations for 4G, 5G, and 6G and the heterogeneous networks are shown as dotted lines on the figure). For 40% collusion rate, M1 detects a malicious miner in 83.12 s, 257.3 s & 341.8 s whereas M2's latency is higher than the failure limit. M3 doesn't converge at all at this collusion rate. Based on these results we can see that when 6G is combined with blockchain in a large-scale CAV application scenario, M1 based dPoS consensus can detect malicious activity in few seconds (less than a minute in most cases). Heterogeneous deployment scheme also has very promising results and can be particularly useful for such timely detection.

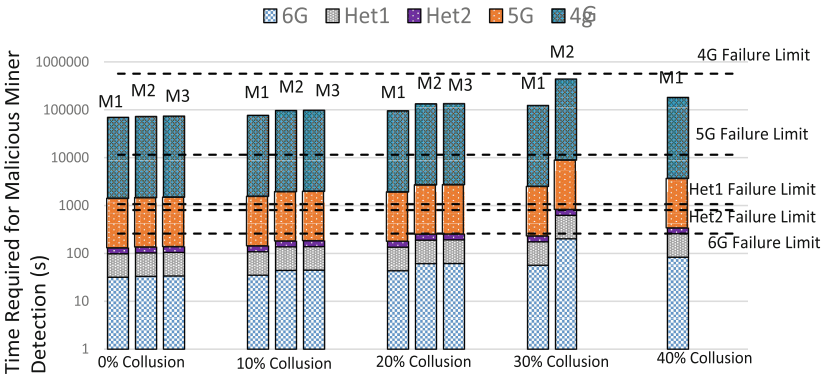

Fig. 3. Orchestrated Adversary Model: Time required to detect malicious miner

5 Conclusion

In this paper, we considered the use of blockchain for a challenging CAV application scenario that has stringent security requirements. We discussed enhanced dPoS consensus and combined it with three different reputation schemes, which are MWSL reputation, beta reputation, and sigmoid-based reputation schemes. To determine malicious miners in the system, we first computed the overall latency of block generation as the sum of communication, computational, information propagation and queuing latency. We then evaluated the performance of this scheme under simple and orchestrated adversary attack models. We did the same analysis for a heterogeneous network, where we validated our results with a 5G/6G hybrid network. Our results indicated that in large scale networks, MWSL reputation based enhanced dPoS scheme can detect orchestrated attacks in few seconds for 6G networks. Heterogeneous deployment schemes also perform relatively well. In comparison, 4G and 5G perform poorly, and might not be suitable for blockchain implementations. Further reduction in detection times may be achieved through less resource intensive consensus algorithms at the expense of some reduction in security performance.

References

1. Gai, F., Wang, B., Deng, W., Peng, W.: Proof of reputation: a reputation-based consensus protocol for peer-to-peer network. In: Pei, J., Manolopoulos, Y., Sadiq, S., Li, J. (eds.) DASFAA 2018. LNCS, vol. 10828, pp. 666–681. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91458-9_41
2. Hassan, N.U., Yuen, C., Niyato, D.: Blockchain technologies for smart energy systems: fundamentals, challenges, and solutions. IEEE Ind. Electron. Mag. **13**(4), 106–118 (2019). https://doi.org/10.1109/MIE.2019.2940335
3. Jerew, O., Blackmore, K.: Estimation of hop count in multi-hop wireless sensor networks with arbitrary node density. Int. J. Wirel. Mob. Comput. **7**(3), 207–216 (2014)

4. Jiang, H., Zhang, Z., Wu, L., Dang, J.: A non-stationary geometry-based scattering vehicle-to-vehicle MIMO channel model. IEEE Commun. Lett. **22**(7), 1510–1513 (2018)
5. Josang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th Bled Electronic Commerce Conference, vol. 5, pp. 2502–2511 (2002)
6. Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D.I., Zhao, J.: Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory. IEEE Trans. Veh. Technol. **68**(3), 2906–2920 (2019)
7. Kang, J., et al.: Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet Things J. **6**(3), 4660–4670 (2018)
8. Khan, A.H., et al.: Blockchain and 6G: the future of secure and ubiquitous communication. IEEE Wirel. Commun. 1–8 (2021). https://doi.org/10.1109/MWC.001.2100255
9. Mollah, M.B., et al.: Blockchain for the internet of vehicles towards intelligent transportation systems: a survey. IEEE Internet Things J. **8**(6), 4157–4185 (2021). https://doi.org/10.1109/JIOT.2020.3028368
10. Parkinson, S., Ward, P., Wilson, K., Miller, J.: Cyber threats facing autonomous and connected vehicles: future challenges. IEEE Trans. Intell. Transp. Syst. **18**(11), 2898–2915 (2017)
11. Ferdous, M.S., Chowdhury, M.J.M., Hoque, M.A., Colman, A.: Blockchain consensus algorithms: a survey. arXiv preprint arXiv:2001.07091 (2020)
12. Singh, M., Kim, S.: Blockchain based intelligent vehicle data sharing framework. arXiv preprint arXiv:1708.09721 (2017)
13. Song, H.: Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane. In: Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 127–132 (2013)
14. Suthar, P., Agarwal, V., Shetty, R.S., Jangam, A.: Migration and interworking between 4G and 5G. In: 2020 IEEE 3rd 5G World Forum (5GWF), pp. 401–406. IEEE (2020)
15. Wan, L., Guo, Z., Chen, X.: Enabling efficient 5G NR and 4G LTE coexistence. IEEE Wirel. Commun. **26**(1), 6–8 (2019)