Proof-of-Communication-Capability Based Authentication in Blockchain-enabled Wireless Autonomous Vehicular Networks

Ali Hussain Khan, Chuadhry Mujeeb Ahmed, Naveed Ul Hassan, and Zartash Afzal Uzmi

***Abstract*—Blockchain technology is finding applications in wireless networks, in particular vehicular networks, for the purposes of establishing trust. A certain set of network resources in terms of communication and computation requirements is also necessary for successful blockchain deployment. Moreover, heterogeneity in wireless networks and changing radio conditions at the physical layer, make it even more challenging to guarantee steady block generation latency. In addition to this, these diverse and complex scenarios give rise to increasing threats to blockchain functionality in safety critical applications such as Autonomous Vehicular (AV) networks. In this work, we study the possible denial of blockchain service attacks where malicious nodes threaten to slow down the block generation latency. We propose a novel *Proof-of-Communication-Capability* (PoCC) authentication framework that acts as a defense against communication capability spoofing over wireless networks. Our PoCC authentication framework utilizes the physical properties such as distance between nodes, channel state information (CSI), and communication puzzle latency to establish the communication capabilities of nodes in the wireless network. Results from a simulated AV network under three different variations of the proposed PoCC framework are encouraging and demonstrate that such attacks can be effectively mitigated.**

I. Introduction

With the rapid development in communication and computing technologies, decentralized Autonomous Vehicular (AVs) networks of self-driving cars and drones are becoming a reality. These connected vehicles have on-board sensing, actuation and control capabilities [1]. Frequent data sharing and time-critical cooperative decision making is often required in AV networks, which is enabled by the advances in software & hardware stack as well as fast wireless communication links. At the same time, due to dynamic and mobile network environment along with decentralization, data sharing and autonomous decision making, threat surface of AV networks is also increased [2], [3].

Recently, blockchain has emerged as a solution to security and trust issues in decentralized networks. Because of data replication at multiple nodes, it also provides protection from a single point-of-failure in these networks. Blockchain relies on a network-wide consensus that has to be achieved through frequent communication within the nodes in order to establish trust on transactions (data sharing) that occur among various nodes of the system [4]. Blockchain is being considered as an enabler of trustless environment in numerous application domains such as Internet of Things (IoT) [5], smart grids [6], and future cellular networks [7]. Connected AV networks are decentralized and highly mobile cyber-physical systems that can also benefit from blockchain technology. In these networks, the nodes communicate over wireless channels as a natural choice. Blockchain deployment on wireless channels requires relatively more communication resources as compared to the wired channels [8].

A. Blockchain dependence on network resources and network heterogeneity

Blockchain technology is resource-intensive and there is an underlying assumption of sufficient resource provision both in terms of communication and computation capabilities to keep up with the desired block generation rate.

Blockchain dependency on communication resources: Blockchain is driven by consensus which is an agreement between the nodes participating in the network on the state of the network. This consensus is highly dependent on the communication resources available to various nodes in the network. Better communication resources would lead to decreased latency whereas degraded communication resources would result in increased latency [7]. Increased network latency would also disrupt the network convergence to a global state. In probabilistic blockchain consensus mechanisms like Proof-of-Work (PoW), increased network latency would result in network forks that would also prolong network convergence and transaction finality. Since only one of the forks become the longest chain, the work done on the other forks would also be wasted [9]. In deterministic and communication intensive distributed consensus mechanisms like practical Byzantine Fault Tolerance (PBFT), the effect of network latency is compounded because votes are exchanged and tallied within the network. With a multiple broadcast based consensus (pre-prepare, prepare and commit transactions), increasing the network latency will increase the overall consensus latency multifold while severely affecting the scalability of the system [10].

Blockchain technology and network heterogeneity: Network heterogeneity also plays a very important role in blockchain communication latency and ultimately impact con-

A. H. Khan, N. U. Hassan and Z. A. Uzmi are with the Department of Electrical Engineering, Lahore University of Management Sciences (LUMS), 54792 - Lahore, Pakistan. (Emails: 18060048@lums.edu.pk, naveed.hassan@lums.edu.pk, zartash@lums.edu.pk).

C. M. Ahmed is with the department of Computer And Information Sciences, University of Strathclyde, Glasgow. (Email: mujeeb.ahmed@strath.ac.uk).

This is a peer-reviewed, accepted author manuscript of the following article: Khan, A. H., Ahmed, C. M., Hassan, N. U., & Uzmi, Z. A. (2022). Proof-of-communication-capability based authentication in blockchain-enabled wireless autonomous vehicular networks. In 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring) IEEE. https://doi.org/10.1109/vtc2022-spring54318.2022.9860891

sensus time. The more resource constrained and unstable connected nodes present in the network, the higher is the network latency and the longer it will take to achieve consensus in the network. In [11], the authors implement blockchain in a heterogeneous network. They observed that the network latency of blockchain running on a dedicated ethernet connection is much lower than that on a dedicated Wi-Fi network. For a network with coexisting nodes, the latency will be defined by the ratio of both types of nodes. A similar trend can be observed for cellular networks where a dedicated 4G connected network will have much higher latency than a dedicated 5G connected network and a network with both types of nodes will have latency based on the ratio of both types of network nodes coexisting in the same network [7].

Communication link quality is stable in wired networks. Therefore, blockchain technology has been shown to work reasonably well and block generation latency can also be guaranteed on such networks [8]. On the other hand, the quality of wireless communication links varies due to interference and changing environmental conditions. In the case of mobile nodes as we have in connected AV networks, disconnections and communication link quality degradation can occur more frequently due to handovers and changing distance between the transmitters and receivers [12]. On top of these inherent link degradations and disruptions, malicious nodes in the wireless network may also try to undermine the network performance by deliberately slowing down or denying the blockchain functionality by spoofing their communication capabilities. Blockchain deployed on wireless networks is more vulnerable to communication capability spoofing attack such as Denial-of-Service (DoS) where adversaries would tend to slow/halt the blockchain consensus and hence the block generation rate for malicious gains. Moreover, due to unstable and changing wireless link quality, identifying such malicious nodes also becomes challenging.

B. Our Contributions

In this work, we consider communication capability spoofing attacks in AV network scenario comprising of Connected Autonomous Vehicles (CAVhs) and Roadside Units (RSUs). CAVhs generate and share data while RSUs with higher computational resources are responsible for consensus and block generation. All the communication links are assumed to be wireless. CAVhs and RSUs can leave or join the network and RSUs can turn malicious. We consider four cases of malicious behavior called Network-Upgrade-Join (NU-Join), Network-Upgrade-Run (ND-Run), Network-Downgrade-Join (ND-Join), and Network-Downgrade-Run (ND-Run). In NU-Join (ND-Join), we consider that an RSU will report enhanced (degraded) communication capabilities at the time of joining the network, while in NU-Run (ND-Run) we assume that an RSU will try to report enhanced (degraded) communication capabilities once is part of the network.

We discuss mitigation against these attacks. For run time spoofing, we discuss some ideas that form the basis for their detection. To counter communication capability spoofing attacks at the time of joining the network, we develop an authentication framework called Proof-of-Communication-Capability (PoCC) where the capabilities of a joining RSU will be verified by the network (through consensus) and stored on the blockchain. PoCC framework takes into account the distance between nodes, quality of communication link and latency of communication puzzle. We develop three variants of PoCC and determine their effectiveness in detecting spoofing attacks through simulations. The best variant is able to mitigate almost all of the attacks.

The rest of the paper is organized as follows. In section II, we discuss the system model and the four attack scenarios. In Section-III, we discuss our PoCC framework and the three variants. In Section-IV, we present the simulation results and finally, we conclude the paper in Section-V.

II. System Model and Attack Scenarios

In this section, we present our system model and the communication capability spoofing attack scenarios.

A. System Model

We consider an AV network where communication occurs over wireless links. The network comprises of a trusted authority (TA), large number of static/mobile CAVhs and RSUs. We assume that secure and timely data sharing is the most important requirement in this network for coordinated/cooperative decision making. For ensuring security, we assume that blockchain technology is employed and all the RSUs in the network can participate in the consensus mechanism. As we want to focus on communication capabilities, we assume that a non-PoW type consensus mechanism such as delegated Proof-of-State (dPoS) or a variant of PBFT is used in the network. Consensus mechanism is therefore a communication-intensive process. To summarize, we have the following nodes in the network.

TA: It deals with registration and authentication of the participating entities in the system. We assume that it issues the public-private key pairs as well as digital certificates to the registered nodes. In addition to this, we assume that TA is also responsible for initiating the PoCC authentication process when a new RSU wants to join the system.

RSUs: RSUs are the infrastructure elements which are deployed on the sides of the roads. They are assumed to have high computational and storage capabilities due to which they serve as the consensus nodes. RSUs store complete copy of blockchain and also update it after every cycle of transaction and verification.

CAVhs: CAVhs are the nodes which use the services provided by the AV network to safely maneuver on the roads. CAVhs share data with each other and rely on blockchain for the security of their shared data.

In our previous work [7], we found that in a large-scale AV network, block generation latency is several hundred seconds when communication speeds are slow (like 4G) as compared to only few seconds when communication speeds are fast (like 5G). Therefore, the propagation of messages and

block generation in AV network is highly dependent on the communication speeds of the consensus nodes. Hence, for successful blockchain functioning, it becomes important to verify the communication capabilities of the consensus nodes in the network.

B. Attack Scenarios

Recently, capability attacks have been executed and reported on 4G/5G devices [13]. The attackers downgrade or upgrade to a particular protocol in order to reduce security capabilities of the device to gain illegal access to it or to obtain undue monetary advantages. In our AV network, capability spoofing attacks by RSUs can severely impact blockchain functionality. In the following we elaborate all such attacks.

1) NU-Join: In NU-Join, the malicious RSU reports upgraded wireless communication capabilities while joining the network. Slow nodes are discouraged from joining because they can prolong the consensus. Therefore, by falsely reporting superior communication capabilities, RSUs may try to sneak into the network. For this case, there could be some detection challenges due to backward compatibility and channel variations due to shadowing and fading. However, this spoofing can be easily checked by issuing a communication challenge to the joining node according to its claimed capabilities. Since the joining node cannot alter the physical nature of its resources and it also does not know the communication challenge in advance, therefore, it is highly likely that it would fail the test and hence the spoofing would be detected.

2) NU-Run: In NU-Run, RSU turns malicious after joining the network and starts reporting upgraded wireless communication capabilities. In several consensus mechanisms, the participating nodes are given incentives based on the resources committed in each round. For example, in [14] various smart contracts are designed according to the capabilities of the nodes. Hence, the motivation of the malicious node is to claim higher incentives. However, there is a high chance of detecting this attack because the spoofing node will most likely fail to solve the smart contract designed for the faster nodes. In addition to this, TA can also ask the nodes to go through a re-verification of capabilities. Incentives can also be redesigned keeping in consideration the changing quality of wireless channels. This is an interesting topic for future research.

3) ND-Join: In ND-Join, the malicious RSU reports downgraded wireless communication capabilities while joining the network. In this case, the intention of the malicious node is to prolong the consensus in the system and thus deny the usage of blockchain services, i.e., Denial of Service attack, similar to *Slowloris Attack* [15] on application layer. This is the most interesting attack because the malicious node can easily complete any issued communication task according to its reported capabilities. As discussed before, the latency of adding a block to the chain could increase several hundred times in an AV network dominated by nodes with degraded communication capabilities. If a node reputation-based system is used in the network, the chances of malicious node staying for longer in the system and causing more damage would further increase because reputations are usually updated when the generated block becomes available on blockchain for auditing [14]. A discouraging factor against this attack would be the low percentage of slow nodes allowed in the system in the first place. Node hardware fingerprint could also contribute to detecting false reported capabilities. Even if the node becomes a part of the system with spoofed wireless communication capabilities, it could be detected based on other measurements like power consumption and sleep mode power consumption.

4) ND-Run: In ND-Run, RSU turns malicious after joining the network and starts reporting degraded wireless communication capabilities. Here, the motivation again is to slow down the consensus and sabotage the system for illegal gains. A node may also downgrade itself during the run time to increase the chances of success as it might not be allowed to do ND-Join because of the cap on the number of slow nodes. This type of spoofing can be detected by measuring anomalies in the communication channel like received signal strength. For example, if the node starts transmitting over a lower frequency band and also decrease the transmission bandwidth, it would clearly indicate the downgrading of wireless communication capabilities. The features of the physical channels and capabilities of the devices make the central point of this work.

III. Proof-of-Capability Based Authentication in Blockchain

In this work, we assume that RSUs can get compromised or turn malicious. We assume that the process of initialization of authentication/security-association among nodes is done on a secure channel. So, the information exchanged between the joining nodes and the TA could not get compromised and the nodes transmit the information to the TA as they intend. In the following, we explain *Proof of Communication Capability* (PoCC) authentication framework. The detailed PoCC mechanism is also shown in Fig. 1.

A. Consensus Protocol

The consensus protocol for the PoCC based authentication is outlined in the following steps.

Step 1: The new RSU that intends to join the blockchain network as consensus node sends a request to the TA for registration. The TA queries the ID of the RSU and issues it a short-lived identity that is subject to the the result of the PoCC based consensus in the system. The TA forwards a request to the current leader of the system.

Step 2: The leader upon receiving the request to verify the capabilities of the joining node designs distinct communication tasks of the same size and sends them to n verifier nodes (RSUs) within a particular radius. Here, the task is that the verifier node transmits a file of a particular size to the prover node. The prover transmits the file back to the verifiers. The verifier calculates the tuple (d_n, CSI_n, t_n) where d_n is the distance from the verifier node measured using distance measurement techniques, CSI_n is the channel state information of the channel between prover and verifier (indicator of channel

quality) and t_n is the one way latency measurement of file transmission measured as half of the round trip time from prover to verifier. Tasks are given to the joining node in turn by each of the verifiers and the prover responds to each of them. The results are broadcast to the whole network. Additionally, the leader designs smart contracts and sends them to all consensus nodes, which will solve them subject to their verification of capabilities of the joining node.

Step 3: All the consensus nodes receive the results of all the communication tasks from the verifier nodes. Then the consensus nodes order the received tuples in ascending order of distances. Each consensus node checks the validity of the information. As we are considering wireless links, there is a tolerance for the verification that could be considered anomalous (explained in the next subsection). Based on these reports, the consensus nodes decide if the reported capabilities are authentic and solve the smart contract and send the result to the leader.

Step 4: If the leader receives verification from 2/3 of the system, it will consider the capabilities of the joining node to be established. The leader forwards this result to the TA which will complete the registration process of the joining node. If the system has agreed on its capabilities, it is considered as a legitimate consensus node. Otherwise, the node is flagged and denied entry into the system. Based on agreement, the node is issued a public-private key pair and deemed as a legitimate node in the system. Now, after outlining the core features of PoCC framework, we are proposing spoofing detection algorithm based on PoCC in the following.

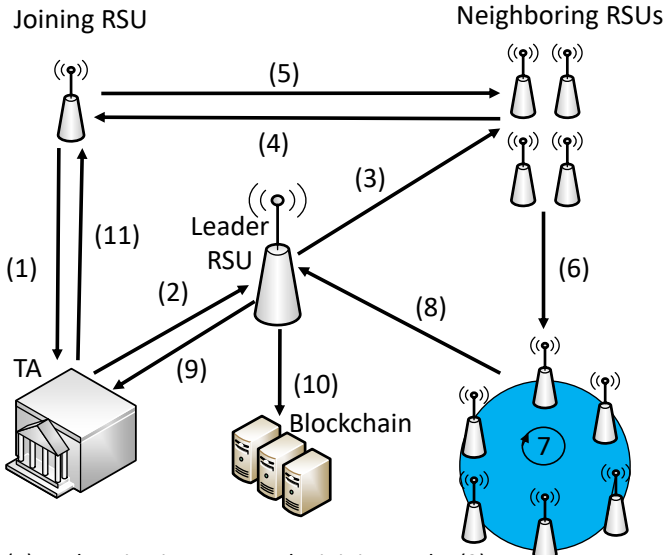

Fig. 1. Information flow of PoCC based consensus protocol.

No.	d_n	t_n	Decision
1	low	low	No Attack
2	low	high	if $CSI_n == bad$:No Attack, else: Attack
3	high	low	if $CSI_n == good$:No Attack, else: Attack
4	high	high	No Attack

TABLE I
V1(CMP(d_n, t_n)): COMPARING t_n WITH d_n TO MAKE A DECISION ABOUT SPOOFING ATTACK.

B. Capability Spoofing Detection Algorithm

In this subsection, we discuss the capability spoofing detection algorithm during run time. We will discuss three variations named Variation 1 (V1(CMP(d_n,t_n)) or simply V1), Variation 2 (V2(CMP(d_n,($t_{n,low}, t_{n,high}$))) or simply V2) and Variation 3 (V3($t_{n,low} = f(d_{n,low}), t_{n,high} = f(d_{n,high})$)) or simply V3) of the algorithm as follows:

V1(CMP(d_n,t_n)): Considering the two variables d_n, t_n, we state all the possible combinations of high/low states in the Table I along with the respective detection of attack. From the reported values of d_n and t_n, if d_n is low and t_n is low, then there is no issue in the measurement and the measurement is considered authentic. If d_n is low but t_n is high, then it checks if CSI_n is also bad. If it is, then the measurement is considered authentic and the high t_n is attributed to the bad radio conditions. Otherwise, if CSI_n is good, then the measurement is considered to be spoofed and is flagged. If d_n is high but t_n is low, then check CSI_n information. If CSI_n is good, then the measurement is considered authentic and the low t_n is attributed to the good radio conditions. Otherwise, the measurement is considered to be spoofed and is flagged. If d_n is high and t_n is also high, then the measurement is considered authentic. However, this set of conditions might fail as explained in the following.

V2(CMP(d_n,($t_{n,low}, t_{n,high}$)): Consider the scenario in the second row of the Table I where d_n is low and t_n is high and for a bad CSI, it is declared to be no-attack. An obvious limitation of the above variation is that every time, bad CSI_n is reported, the high value of t_n is ignored and attributed to bad CSI_n. Adversary nodes could use this to their advantage and downgrade their capabilities as they would not be detected if their CSI_n is bad. To cater to this limitation, we present V2, where a comparison(CMP) is done with the high value of t_n for bad CSI_n to the upper and lower limits of t_n. If the value of CSI_n is bad and t_n is within the limits, the bad t_n is attributed to bad CSI_n. Otherwise, the node is spoofing its capabilities. This is summarised in the Figure 3 with the quantitative numbers from the experimental results discussed in the results section.

V3($t_{n,low} = f(d_{n,low}), t_{n,high} = f(d_{n,high})$): In V2 there is still a limitation, where the 4G nodes located closer to the receiver will report values between the above defined threshold. Therefore, the 5G node spoofing to be a 4G node will be ignored as legitimate 4G nodes are also reporting those values. To cater to this problem, we will define t_n bounds for different d_n ranges, making those limits a function of

distance. So
it will be fla
that particul
detection ac

C. Threat M

Adversari
Some of th
detect and t
sary could t
the verifiers
node issuing
try to spoof
advance in c
to the rando

One dime
neighboring
can report s
relies on th
neighboring
the joining n
results will
gation. The
possible atta
tive. In this work we will consider the ND-Run attack. Here, the attacker tries to downgrade its capabilities while running the blockchain. We will also consider the attack detection based on the above designed detection algorithm against all variations. The attack and attacker model explained in this section is integrated in the simulation setup explained in the following.

IV. Results

In this section, we provide simulation results based on the above discussion. First, we will consider the impact of heterogeneous nodes on the latency of block generation and addition to the blockchain. We consider a scenario where 1000 RSUs are deployed in a 150 sq. km area and 1000 vehicles in the system which are sharing the data among each other, using the blockchain services to get the data validated. We implement a data sharing and consensus procedure similar to [14] and use the other simulation parameters as used by [7]. We consider a deployment where all nodes are operating over 5G network. To consider the impact of heterogeneity, we introduce 25% 4G nodes to the system with the rest 75% of 5G nodes to observe their impact on the block generation time. Next, in two separate variations, 50% and 100% 4G nodes are considered, respectively, and their impact on block generation time is studied. The data rates for 4G and 5G nodes are considered to be 10 Mbps and 500 Mbps respectively. The results for this case are shown in Fig 2. We can see that for a completely 4G network, the latency of block convergence is 35.9205s and for a completely 5G network, the latency is 0.7674s. For a small level of heterogeneity where only 25% nodes are considered to be 4G nodes, the latency is increased by multifold i.e., to 4.6268s. If we further increase

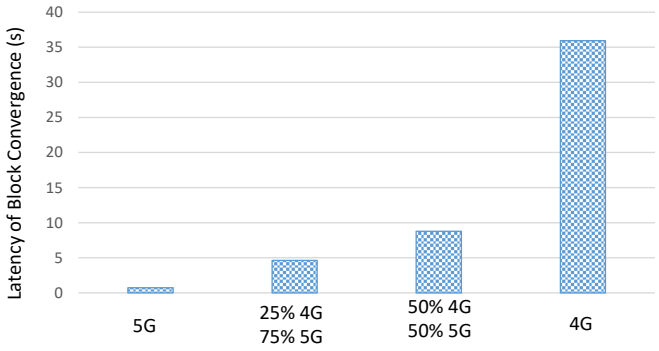

Fig. 2. Block convergence latency of different network node distributions.

Next, we simulate the results for the designed detection algorithm. We consider that a node spoofs its capabilities once it has entered the system as a 5G node. We will consider that there are 20 nodes that will frequently communicate with the prover node and query CSI_n, d_n and t_n. The d_n values vary between 0 and 1000m. We consider a communication model where there is distance-dependent path loss and the channel follows Rayleigh fading. The fading variable α varies between 0 and 1 which is sampled from Rayleigh distribution which is the indicator of good or bad CSI_n. We tested the designed algorithm for the CSI_n cutoff at α values of 0.3, 0.5 and 0.7, where good CSI_n is above the cutoff and bad CSI_n is below the cutoff. We assume that the verifier sends a file of 125 KB to the receiver to prove its capabilities. For V1, we put latency cutoffs for α values of 0.3, 0.5 and 0.7 to be 0.0108s, 0.008s and 0.007s. The range for latency values for V2 are 0.0108s-0.08s, 0.008-0.08s and 0.007-0.08s. Figure 3 shows the idea visually for the case of V1 and V2. The latency cutoffs for V3 are given in Table II and the detection accuracy of the designed algorithm variations are given in Fig. 4.

From Fig. 4, we can see that generally, V3 performs the best and V1 performs the worst. As discussed is III-B, V1 is the most lenient variation and only puts one bound on t_n for

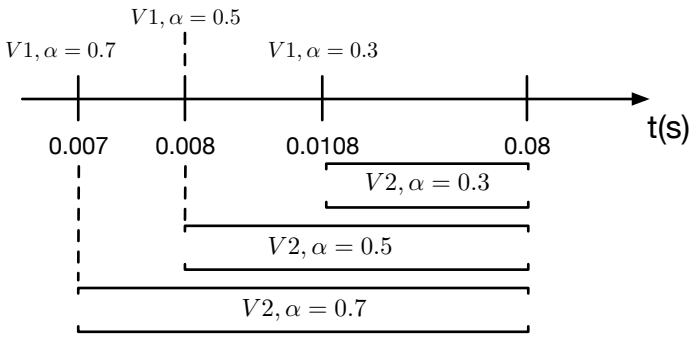

Fig. 3. V1 and V2 detection functions. V1(CMP(d_n,t_n)) compares the d_n with given point of latency time t_n for each α fading coefficient. V2(CMP(d_n,($t_{n,low}, t_{n,high}$)), compares with a range of t_n for each α, as shown here.

TABLE II
LATENCY RAGES AT DIFFERENT α AND DISTANCE RANGES FOR BAD CSI FOR 125 KB FILE SIZE

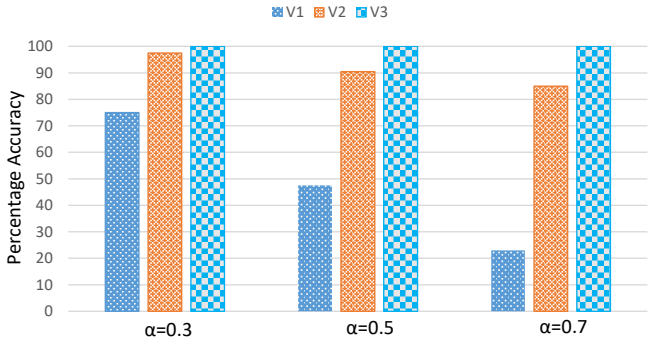

Fig. 4. Accuracy of detection algorithm for three fading coefficients. V3 ($t_{n,low} = f(d_{n,low}), t_{n,high} = f(d_{n,high})$) performs the best as it is hard for an attacker to defy the physical properties, given a precise mapping of distance to latency.

the classification. Whereas, V2 puts an upper and lower bound on t_n, so it performs better than V1. And V3 bounds both t_n and d_n. Therefore, it gives the best accuracy. For larger values of α, the accuracy of the different variations of the algorithm decreases and vice versa. This is due to the inherent design of the algorithm. As α increases, the range of low CSI_n increases and therefore, there is a higher chance that the spoofing will be ignored. Specifically, for α =0.3, the accuracy values for V1, V2 and V3 are 75%, 97.5% and 100%. For α =0.5, the accuracy values are 47.5%, 90.5% and 100% and for V3, the accuracy values are 23%, 85% and 100% respectively.

V. CONCLUSION

Proof-of-Communication-Capability (PoCC) authentication framework based on the physical properties of the communication channels and the nodes is proposed in the paper. The framework is developed for effective detection of malicious nodes that communicate over wireless links and spoof their communication capabilities in order to sabotage blockchain performance deployed in AV network. It is concluded that more heterogeneous a network is, more it is susceptible to slow convergence of blockchain consensus and to malicious node degradation attacks. The proposed PoCC framework provided a set of consensus rules that helped us detect the malicious nodes and since the PoCC is based on the physical features, e.g., location, channel state information, propagation time, it is not easy for an adversary to evade the detection mechanism. *Proof-of-Communication-Capability* framework shall prove to be an excellent frame of reference in future implementations of blockchain in resource constrained wireless communication systems.

REFERENCES

[1] A. Chowdhury, G. Karmakar, J. Kamruzzaman, A. Jolfaei, and R. Das, "Attacks on self-driving cars and their countermeasures: A survey," *IEEE Access*, vol. 8, pp. 207 308–207 342, 2020.

[2] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 11, pp. 2898–2915, 2017.

[3] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[4] L. Yue, H. Junqin, Q. Shengzhi, and W. Ruijin, "Big data model of security sharing based on blockchain," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, 2017, pp. 117–121.

[5] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2018.

[6] N. U. Hassan, C. Yuen, and D. Niyato, "Blockchain technologies for smart energy systems: Fundamentals, challenges, and solutions," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 106–118, 2019.

[7] A. H. Khan, N. U. Hassan, C. Yuen, J. Zhao, D. Niyato, Y. Zhang, and H. V. Poor, "Blockchain and 6G: The future of secure and ubiquitous communication," *IEEE Wireless Communications*, pp. 1–8, 2021.

[8] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How much communication resource is needed to run a wireless blockchain network?" *arXiv preprint arXiv:2101.10852*, 2021.

[9] L. Wan, D. Eyers, and H. Zhang, "Evaluating the impact of network latency on the safety of blockchain transactions," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 194–201.

[10] Y. Meshcheryakov, A. Melman, O. Evsutin, V. Morozov, and Y. Koucheryavy, "On performance of PBFT blockchain consensus algorithm for IoT-applications with constrained devices," *IEEE Access*, 2021.

[11] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: performance of resource-constrained IoT networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174–183, 2020.

[12] H. Balbi, D. Passos, R. C. Carrano, L. Magalhães, and C. Albuquerque, "A case study of association instability in dense IEEE 802.11 networks," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[13] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New vulnerabilities in 4G and 5G cellular access network protocols: Exposing device capabilities," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 221–231. [Online]. Available: https://doi.org/10.1145/3317549.3319728

[14] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[15] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 318–322.