A Scalable Protocol for Driving Trust Management in Internet of Vehicles with Blockchain

Uzair Javaid, *Student Member, IEEE*, Muhammad Naveed Aman, *Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Recent developments in IoT have facilitated advancements in Internet of Vehicles (IoV) with autonomous vehicles and roadside infrastructure as its key components. IoV aims to provide new innovative services for different modes of transport with adaptive traffic management and enables vehicles to broadcast messages to improve traffic safety and efficiency. However, due to non-trusted environments, it is difficult for vehicles to evaluate the credibility of the messages that they receive. Therefore, trust establishment in IoV is a key security concern that is constantly limited by scalability challenges. This paper proposes a blockchain based protocol for IoV using smart contracts, Physical Unclonable Functions (PUF), certificates, and a dynamic proof-of-work (dPoW) consensus algorithm. The blockchain, in conjunction with contracts, provides a secure framework for registering trusted vehicles and blocking malicious ones. PUFs are used to assign a unique identity to each vehicle via which trust is established. Certificates are issued by roadside units which preserve the privacy of vehicles, whereas the dPoW consensus allows the protocol to scale according to the incoming traffic generated by the vehicles. To demonstrate the feasibility and scalability of the proposed protocol, security and performance analyses are presented. A case study is also discussed along with a comparative analysis, which confirms that our protocol can provide superior decentralized trust management for IoV.**

***Index Terms*—IoV, blockchain, trust management**

I. Introduction

The number of vehicles in vehicular networks is constantly rising and may cross two billion within the next 10-20 years [1]. The Internet of Vehicles (IoV) is critical for realizing this number and to facilitate next-generation intelligent transportation systems (ITS) and related technologies. One of the primary IoV objectives is to enhance traffic efficiency and improve the safety of vehicles, its passengers, and the pedestrians alike. According to World Health Organization (WHO), traffic accidents account for approximately 1.25 million deaths every year. Therefore, trust management protocols in IoV are needed to guarantee road safety measures.

IoV is a distributed network of vehicles and roadside infrastructures that is capable of managing data generated by vehicles and their associated networks. It is also envisioned that vehicles in the IoV would have the capability to communicate in real-time with their human drivers, pedestrians, other vehicles, roadside infrastructure, and fleet management systems [2], [3]. Most of the existing IoV trust management protocols do not provide robust security measures and lack a reliable system for securely registering and revoking vehicle registrations [4]. One of the main challenges in developing secure IoV protocols is scalability which requires a protocol to be efficient in terms of communication overhead, thereby enabling its large-scale deployment. Given the increasing number of vehicles and stricter latency requirements, the security overhead needs to be minimized. Thus, in this paper, we use Physical Unclonable Functions (PUF) as the root of trust for the proposed protocol. PUFs have certain features which make it an attractive choice for IoV such as physical security, high throughput with low energy and silicon area footprints, low-cost, simple construction, and unclonablity [5].

Vehicular networks typically depend on decentralized resources, i.e., vehicles communicate with the infrastructure via roadside units (RSU) distributed over a large geographical area. Therefore, centralized systems are not suitable for establishing and managing trust in vehicular networks [6]. However, most of the existing IoV applications rely on centralized models. To address this issue, we use blockchain to develop a decentralized protocol with no central authority, which provides a secure way of managing vehicle registrations. A blockchain is a globally distributed ledger that can be public, private, or semi-private. Many applications have adopted blockchain to facilitate distributed trust management and reliance in cyber-physical systems due to its properties of decentralization, immutability, transparency, and fault-tolerance [7]. However, running blockchain mining processes (e.g., performing proof-of-work (PoW)) while supporting increasingly intelligent applications and their operations requires huge computing and storage resources. Therefore, different avenues (e.g., real-time processing, resource-intensive applications, mining, and consensus algorithms) have been identified to address the scalability challenge of blockchain based solutions [7].

A. The four-way trade-off of blockchain

The CAP (Consistency, Availability, tolerance to network Partitions) theorem of distributed systems states [8]: “A robust distributed system can only simultaneously provide two out of the three properties”. Similarly, scalability in a blockchain based system can be considered as a challenge with a four-way trade-off. Figure 1 shows an overview of a IoV-blockchain

This research was supported in part by Singapore Ministry of Education Academic Research Fund Tier 1 (R-263-000-D62-114).

U. Javaid and B. Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583. Email: uzair.javaid@u.nus.edu, bsikdar@nus.edu.sg.

M. N. Aman is with the Department of Computer Science, National University of Singapore, 13 Computing Drive, Singapore 117417. Email: naveed@comp.nus.edu.sg.

model where the yellow spaces represent road lanes with vehicles that are securely communicating (depicted with a magenta lock icon) with the blockchain via a RSU. The four-way trade-off is represented by pink highlighted circles in the figure, which involves the following quintessential factors:

1) **Scalability**: It is the quantitative measure of the ability of a blockchain to handle and process transactions such that a blockchain should be able to handle high volumes of transactions supported by a wide range of applications [9]. Scalability in the proposed protocol is achieved at two levels: (i) lower communication overhead and (ii) a dynamic consensus algorithm that adapts to the incoming traffic rate.

2) **Decentralization**: It is the distribution of control/resources in a blockchain which allows it to achieve different objectives, i.e., open participation, censorship resistance, immunity from certain attacks, and elimination of single point of failure [10]. The proposed protocol realizes decentralization through multiple and geographically distributed server, miner, and RSU instances as illustrated in Figure 1.

3) **Latency**: It is the time taken for a transaction to be verified/confirmed and added to a block in a blockchain, thereby becoming irreversible [11]. It can be measured both in terms of time-to-finality (TiF) or authentication delay. The ultra-high throughput of PUFs [12] combined with fast verification times of the dynamic consensus algorithm translate into low latency for the proposed protocol.

4) **Security**: It is the guarantee for the immutability of a ledger in a blockchain and the data it contains, which is arguably reflected by its robustness and resistance to attacks such as 51%, Sybil, and DDoS [13]. Moreover, by defining the root of trust for vehicles via PUFs, foolproof security is ensured [14], [15], [16]. Note that the pink circle around the blockchain instance in Figure 1 represents the challenges of scalability and guaranteeing high security fidelity.

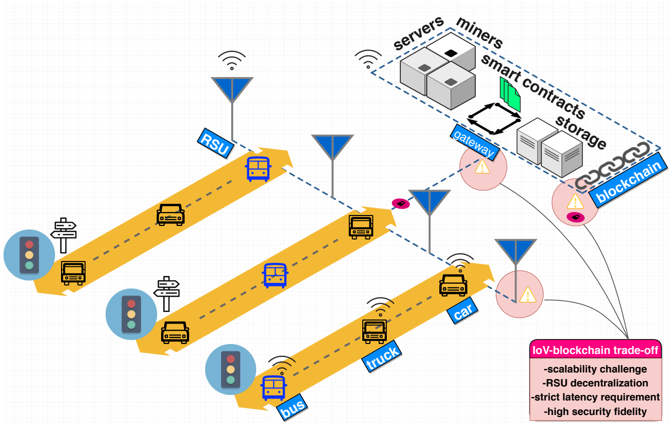


Fig. 1: A IoV-blockchain model with the four-way trade-off.

Many of the existing blockchain based systems only achieve part of these four properties and compromise on the others. For example, blockchain systems that are permissionless by design and use PoW consensus algorithms (e.g., Bitcoin and Ethereum 1.0) can achieve good decentralization with high security fidelity, but suffer from poor scalability and high TiF. Systems that have centralized block production (e.g., Cardano and EOS) tend to achieve high scalability by sacrificing the decentralization of miners. Moreover, multi-chain systems (e.g., Cosmos and AION) achieve scalability with good decentralization and lower TiF but at the expense of undertaking additional attack risks. However, to design a robust distributed system, all four trade-offs need to be considered. Therefore, this paper focuses on designing a decentralized trust management protocol for IoV while addressing the four-way trade-off.

This paper proposes a blockchain based protocol for driving trust management in IoV. It uses smart contracts with PUF, certificates, and a dynamic proof-of-work (dPoW) consensus algorithm. The blockchain and smart contracts provide a secure way of managing vehicle registrations. PUFs are used to assign a unique identity to each vehicle. Certificates are issued to vehicles by RSUs which preserve their privacy. Moreover, the dPoW consensus allows the protocol to scale according to the incoming traffic generated by the vehicles. Thus, the key contributions of this paper can be summarized as follows:

(i) A decentralized and scalable protocol for driving trust management in IoV.
(ii) A reliable framework for managing vehicle registrations via the blockchain public key infrastructure (PKI) features and smart contracts.
(iii) A hardware primitive based assignment of unique vehicle identities via PUFs as well as preservation of their privacy by issuing them certificates.
(iv) A dynamic consensus algorithm that can scale and change its operation according to the incoming traffic rate generated by the vehicles.
(v) A rigorous performance analysis of the protocol while addressing the four-way trade-off of blockchain.
(vi) A case study to evaluate the protocol dynamics along with a comprehensive comparative analysis with a state-of-the-art IoV trust management protocol.

The rest of the paper is structured as follows: Section II discusses the related work. Section III explains the protocol and its operation while Section IV presents its security analysis. Section V describes its implementation. Section VI presents the performance analysis while Section VII presents a case study. Finally, Section VIII concludes the paper.

II. Literature Review

Traditional centralized system architectures for vehicular ad-hoc networks (VANETs) find it difficult to cope with the rising complexity of ITS applications. The rapid growth of IoV networks has presented significant challenges for large data storage, trust management, and information security [29].

The authors in [6] propose a dynamic distributed trust model to establish a trust relationship between vehicles in VANETs. The trust model is based on the monitoring of the instantaneous behavior of vehicles that filters out malicious and selfish vehicles. A job market signaling scheme for trust management in VANETs is proposed in [17]. This scheme is based on allocating credits to vehicles and securely managing these credits. To identify and remove malicious nodes, the scheme sets the cost of sending and receiving packets using

TABLE I: Summary of existing trust management protocols.

Feature	[6], [17]	[18]	[19], [20]	[21], [4]	[22], [23], [26]	[24], [25], [27], [28]	Proposed
scalable	✗	✗	✗	✗	✗	✗	✓
decentralized	✓	✓	✓	✓	✓	✗	✓
low latency	✓	✗	✗	✗	✗	✗	✓
security guarantee	✓	✗	✗	✗	✓	✓	✓
unique vehicle ID	✗	✗	✓	✗	✓	✗	✓
physical protection	✗	✗	✗	✗	✗	✗	✓
vehicle privacy	✗	✓	✓	✗	✓	✓	✓
vehicle registration	✓	✓	✓	✗	✗	✗	✓
use of certificates	✗	✗	✓	✗	✗	✗	✓
trust management	✓	✓	✓	✓	✓	✗	✓

- *scalable:* does the protocol address how it scales with more traffic?
- *decentralized:* does the protocol quantify degree of decentralization?
- *low latency:* does the protocol discuss its impact on message latency?
- *security guarantee:* does the protocol justify its security features?
- *unique vehicle ID:* does the protocol assign distinctive IDs to vehicles?
- *physical protection:* does the protocol provide defense against physical attacks on vehicles?
- *vehicle privacy:* does the protocol preserve the privacy of vehicles?
- *vehicle registration:* does the protocol manage vehicle registrations?
- *use of certificates:* does the protocol use certificates for privacy?
- *trust management:* does the protocol establish and manage trust?

economics models of node behavior. Lu et al. [19], [20] propose BARS, a blockchain based trust management system for vehicular networks. They assign credibility scores based on a vehicle's historical data. Nisha et al. [4] also propose an authentication framework for vehicular networks using blockchain. However, these protocols only focus on privacy preservation of vehicles and fail to address their scalability. A crypto trust point (cTp) using blockchain is proposed by Singh et al. [28]. The cTp enables vehicles to securely share data. Similarly, Shrestha et al. [27] discuss a blockchain based message dissemination service for IoV. However, both these solutions do not preserve vehicle privacy. Zhang et al. [25] highlight the amount of data generated in IoV networks and stress on the importance of mobile edge computing to offset resource consumption in blockchain based vehicular networks. Their solution helps in reducing the blockchain computational overhead but the introduction of edge computing does not make it truly decentralized. Furthermore, the authors in [26] propose Trust Bit, a reward-based vehicle communication mechanism. They use blockchain with a unique crypto ID that is issued by the vehicle owner for safe communication.

The authors in [24] introduce a secure platform for data sharing and storage in vehicular networks using a consortium blockchain. However, the use of such a blockchain results in additional overhead and poor scalability. The authors of [18] discuss a blockchain and software-defined networking approach for securing vehicular social networks (VSNs). Their approach makes the vehicular network programmable, virtualized, and partitionable, while the blockchain enables certification of transactions and preserves data integrity. The authors in [30] propose PoolCoin, which is a distributed trust model for reputation management of miners in a blockchain. This idea can be extended to study outsourcing of RSU computational load to mining pools or to allow other miners to partake in regional RSU mining pools. Khelifi et al. [23] present an interesting use of blockchain for secure name data networking caching for IoV. Yang et al. [21] propose a blockchain based decentralized trust management protocol for vehicular networks. Lastly, Lei et al. [22] discuss dynamic key management for heterogeneous vehicular systems. They use blockchain for their proposed key management protocol.

Most of the existing IoV protocols suffer from scalability. Motivated by this challenge, we design a scalable protocol for effectively decentralizing IoV based networks and providing distributed trust management. A summary is presented in Table I that provides a comparison of the proposed protocol with the existing literature. The features listed in the table are addressed by the proposed protocol in the following way:

i **Scalable:** by using a dynamic consensus to demonstrate scaling of throughput, i.e., transactions per unit time, and reduction of communication overhead over time.

ii **Decentralized:** by quantifying the degree of geographical distribution of blockchain miners (RSUs in this case).

iii **Low latency:** by analysing the impact of blockchain packets on vehicle authentication using PUF and time taken for a transaction to be written in the blockchain.

iv **Security guarantee:** by analysing the 51% attack on blockchain and discussing the lengths of security keys.

v **Unique vehicle ID:** by equipping each vehicle with its own unique PUF.

vi **Physical protection:** by using the unclonablity of PUFs.

vii **Vehicle privacy:** by using blockchain account addresses instead of license plates.

viii **Vehicle registration:** by exploiting the PKI framework of blockchain and using smart contracts.

ix **Use of certificates:** to preserve the privacy of vehicles and reduce communication overhead.

x **Trust management:** by using PUFs to provide root of trust in conjunction with blockchain and smart contracts.

III. The IoV-blockchain Protocol

This section explains the core components of the protocol. A summary of the notations used in this paper is presented in Table II.

A. Network model

Figure 1 illustrates the IoV-blockchain network model which consists of the following main components:

1) Vehicle: Depicted with buses, trucks, and cars, this represents the main participants of the network, i.e., intelligent vehicles. Each vehicle has its own blockchain account and a pair of public, private keys for encrypted communication.

2) RSUs: These represent the traffic handling system units that provide wireless communication from roadside infrastructure to vehicles. RSUs are considered as the miners of the protocol that host the blockchain and smart contracts.

3) Smart contract: This represents scripts that are computer codes/programs and can run autonomously. The contract script in the proposed protocol is used to ensure trust establishment and is termed as "enforcer". It is a public contract that interacts with RSUs and ensures that the data generated by the vehicles is coming from a trusted origin. Moreover, it enables safe and secure communication between vehicles and the road infrastructure, i.e., the vehicles send blockchain packets to RSUs via the contracts. It is also responsible for storing and reading data from the blockchain ledger.

TABLE II: Summary of notations

Symbol	Remark
PUF	Physical unclonable functions
CRP	challenge-response pair
C^i	PUF challenge for the i-th iteration
R^i	PUF response for challenge C^i
$\|\|$	concatenation operator
cID_V	crypto identity of a vehicle assigned via PUF
ID_S	identity of a blockchain server
MAC	message authentication code
$H(\cdot)$	hashing function (SHA256 variant)
N_1, N_2	nonces generated for verifying PUF response of a vehicle
$\{N_1\}_{R^i}$	nonce N_1 is encrypted using response R^i
dPow	dynamic proof-of-work
δ	operator to represent different levels of dPoW
$g(\cdot)$	Gini coefficient
TiF	time-to-finality
$t_{f,\delta}$	TiF for δ consensus
$t_{i,\delta}$	block interval time for δ consensus
$t_{c,\delta}$	consensus latency for δ consensus
$\mathbb{R}^2$	two-dimensional coordinates
n_m	number of malicious miners
$n_{m,\delta}$	total number of malicious miners
$n_{h,\delta}$	total number of honest miners
pps	packets per second
PDR	packet delivery ratio
Q_n	probability of an attacker defeating an honest chain

4) Server/miner: This represents a setup or a set of setups that can interact with RSUs and vehicles of the network to provide different kinds of services, e.g., deploying the blockhain itself. The server is the host of the blockchain network, i.e., one who initiates a blockchain with the first block but instead of being centralized, servers are decentralized here. The servers naturally are the trusted hosts since they hold the genesis block that is trusted by all other participants of the network. Moreover, they may employ permissioned or permissionless blockchain protocols to enable interactions between them and the network constituents that include and are not limited to collecting data, processing, querying data from and/or writing data to storage devices etc. In this paper, we consider a permissionless blockchain. Moreover, the miners represent the computational resources of RSUs.

5) Storage: This represents the process of reading and/or writing data to storage devices in the blockchain. Different forms of data (json, xml, csv, etc.) can be stored on them, where the storage may be temporary like RAM (random access memory) or permanent like ROM (read only memory).

6) Physical Unclonable Functions: A PUF is a “hardware fingerprint” that can provide semiconductor devices (e.g., microprocessors, integrated circuits, etc.) with unique identities. PUFs exploit variations that naturally occur during the manufacturing process of semiconductor devices. They are generally used in cryptography and for applications with high security requirements. A PUF is commonly characterized by a challenge-response pair (CRP) which can be represented as:

$$R^i = PUF(C^i). \tag{1}$$

where R^i is the response generated by a challenge C^i. Thus, every PUF produces a unique R^i when excited with a C^i [31], [32]. PUFs are used in the proposed protocol to establish the root of trust as well as replace secret keys and passwords.

7) Blockchain: The blockchain instance represents a locally distributed and geographically bounded ledger that works with smart contracts. The use of a regional blockchain is proposed here because global blockchain solutions have latency and message propagation problems. Therefore, a local blockchain is considered for simplification and to provide low latency.

Listing 1: Data structure of a blockchain packet of a vehicle.

```
{
  "vehicle": {
    "id": "PUBLIC_KEY, the address of vehicle account",
    "cID": "the PUF of vehicle",
    "nonce": "PUF response of vehicle",
    "from": "ID of the sender of the transaction",
    "to": "the address of RSU account",
    "body": "data in a transaction",
    "txIndex": "no. of transactions by vehicle",
    "certificate": {
      "issued": "a boolean value for certificate checking",
      "body": {
        "issueTime": "timestamp when certificate issued",
        "expiryTime": "timestamp when certificate expires",
        "vehicleID": "PUBLIC_KEY",
        "vehiclecID": "vehicle PUF",
        "index": "no. of certificates issued"
      }
    }
  }
}
```

B. The protocol operation

The data structure of the blockchain packets used in the proposed protocol is defined in Listing 1. The primary users of the protocol are vehicles, each of which has a blockchain account with a 20-byte address similar to Bitcoin [33] and Ethereum [34], [35] address sizes. The vehicles are embedded with a PUF that gives them an unique crypto ID (cID) [36]. This makes the vehicles immune to physical and impersonation attacks. However, before any interaction can be made, the vehicles first need to register themselves to become users of the network. Thus, the operation of the proposed protocol consists of two phases: a setup phase for registering vehicles and a data transfer phase for communication among them.

1) The setup phase: This phase is initiated by the enforcer contract which enables communication between vehicles and the local blockchain via Algorithm 1, where $certV$ is the certificate issuing function, V represents a set of vehicles, and txV is the data generated by the vehicles. Figure 2 shows the information flow layout where the vehicles interact with the contract, which consequently interacts with the blockchain via RSUs. Before a vehicle can generate data, it has to be registered first. The functions $vehicle.reg(addr.)$ and $vehicle.del(addr.)$ are responsible for registering and deleting vehicles using their blockchain account addresses, respectively. A list of all registered vehicles in the blockchain is maintained in $veh.registry$ while $CRP.veh$ keeps a list of PUF CRPs for the vehicles. Moreover, RSU represents the certificate authority and $cert.registry$ stores all certificates issued to the vehicles. When a vehicle generates data, the enforcer checks if it is registered. A PUF challenge is then

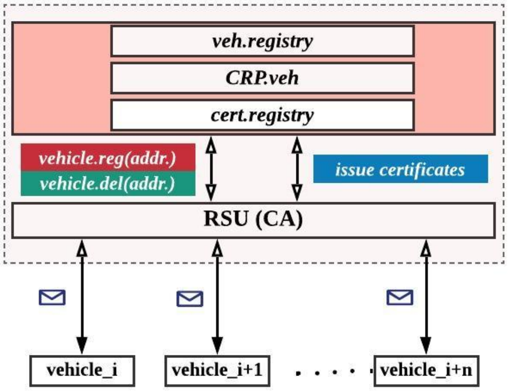


Fig. 2: Flow of a blockchain packet from vehicle to RSU.

sent to the vehicle if it is in the registered list. If it generates a positive response to the challenge, then the link is established successfully between the vehicle and the local blockchain. Finally, after these checks, the RSUs issue a certificate to the vehicle which is then used for its authentication. Thus, after the issuance of the certificate, the vehicle does not have to do these checks again. Note that the certificate issuance is a one-time operation and the certificates issued to the vehicles are valid for the duration they are with the local blockchain. If a vehicle is removed from the registered list, then the certificate issued to it will be revoked and it will have to repeat the setup phase again to register itself and get a certificate.

Algorithm 1: Certificate issuance

```
function: certV
input  : txV
output : authorize/issue/reject
while $txV$ do
    for $n \; in \; v_*, \; v_* \in V \; \forall * = 1, \cdots, v$ do
        if ($v^*[n]$ has certificate) then
            return authorize
        else
            if ($v_*[n]$ is registered in veh.registry) then
                while $txV^*$ do
                    // call PUF CRP protocol
                    if ($PUF^{response}_{v_*[n]}$ == positive) then
                        return issue
                    else
                        return reject
```

2) The data transfer phase: After registration, a vehicle can interact with other vehicles and the blockchain. However, this requires the vehicle to have a certificate, which is issued if it has generated a positive response to its PUF challenge. This paper assumes that each vehicle is equipped with a PUF and the response to a PUF challenge can be obtained in two ways, i.e., either by the vehicle using its PUF or the blockchain operator from a saved copy in its storage. When a vehicle needs to be registered, a CRP for its PUF is already recorded by the operator in the blockchain using the enforcer contract. Thus, each vehicle has its own unique ID along with a unique CRP that is stored in the blockchain. When a vehicle generates data, Algorithm 1 first checks if it has been issued a certificate. If no certificate is issued, it checks if the data is coming from a registered vehicle. If it is registered, the algorithm then checks if its PUF challenge-response is positive. It does so by calling the PUF challenge-response protocol as shown in Figure 3. The steps for this protocol are as follows:

1) A server/miner in the local blockchain with identity ID_S reads the CRP (C^i, R^i) for a vehicle with a crypto fingerprint cID_V and generates a nonce N_1 for it.
2) The server ID_S then sends the nonce N_1 which is encrypted using R^i, i.e., $\{\mathrm{N}_1\}_{R^i}$ and the challenge C^i to the vehicle cID_V in message 1.
3) After obtaining the nonce from ID_S, the vehicle cID_V then obtains the corresponding response R^i for the challenge C^i with the help of its PUF.
4) After obtaining the response R^i, cID_V performs the following steps:
   a) Using R^i as the secret key, obtain N_1 and generate a random nonce N_2.
   b) Verify and validate the message authentication code (MAC) using the parameters in its memory to ensure data integrity.
   c) Once the MAC is verified, it produces a hash: $H(cID_V, data, R^i, N_1, N_2)$.
   d) After generating the hash, it signs the hash with its private key and sends it to ID_S in message 2.
5) Once ID_S receives message 2 from cID_V, it checks and verifies the MAC and the hash using the public key of cID_V. If both are valid, the communication link is successfully established and cID_V is issued a certificate (otherwise, the request is dropped). ID_S then sends an acknowledgement to cID_V in the form of an authentication parameter $I = H(cID_V, N_1, N_2, R^i)$.

It is worth noting here that the crypto fingerprint (cID) is used for secure communication in the IoV-blockchain network. The certificates are used to anonymize the identities of vehicles to preserve their privacy and reduce overhead, i.e., once a certificate is issued to a vehicle, it does not need to go through the PUF challenge.

IV. Security Analysis

This section presents a formal security analysis of the proposed protocol. The following set of assumptions are made and a threat model is described as well.

A. Assumptions

1) Every vehicle is equipped with a PUF.
2) The PUF and a vehicle's microcontroller form a system-on-chip (SoC) and any kind of tampering will render the PUF useless [37], [38].
3) The PUF and microcontroller communicate over a secure channel given the SoC assumption [37], [38].

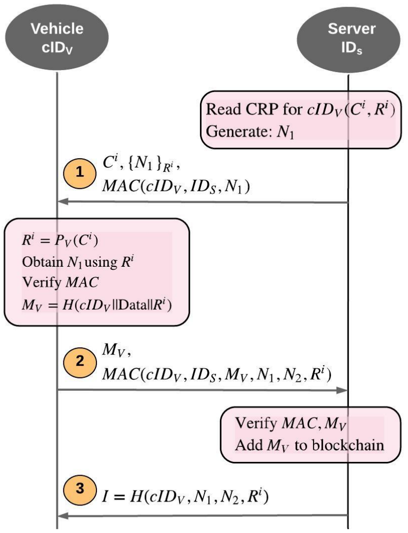


Fig. 3: The PUF challenge-response protocol.

4) The standard assumption regarding a PUF: every PUF is unique and unclonable, i.e., an adversary cannot predict its behavior [39]. A PUF can be modeled as: PUF:$\{0,1\}^{l_1} \rightarrow \{0,1\}^{l_2}$, i.e., a PUF will produce an output of length l_2 when excited with an input of length l_1.

This protocol models PUF security using the following security game, $\text{Exp}^{\text{Sec}}_{PUF,\mathcal{A}}$, between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$:

1) $\mathcal{A}$ randomly chooses a challenge C^i and sends it to $\mathcal{C}$.
2) $\mathcal{C}$ uses the PUF to obtain the response R^i and reveals R^i to $\mathcal{A}$.
3) $\mathcal{C}$ selects a random challenge C^x which has not been used before and obtains the response R^x using the PUF, i.e., $R^x = PUF(C^x)$.
4) $\mathcal{A}$ can query the PUF a polynomial number of times for challenges other than C^x.
5) $\mathcal{A}$ outputs its guess $R^{x'}$ for the challenge C^x.
6) $\mathcal{A}$ wins the game if $R^{x'} = R^x$
   The advantage of the adversary $\mathcal{A}$ in this game can be modeled by $\text{Adv}^{PUF}_{\mathcal{A}} = \texttt{Pr}[R^{x'} = R^x]$.

B. Threat model

A set of vehicles $\mathcal{V} = V_1, V_2, \cdots, V_n$ interacts with the secure blockchain S. The vehicles communicate with S over an unsecured network. At the conclusion of the authentication phase, the entities are either registered in the network or rejected. If authentication is successful, then a vehicle can start to transmit data by interacting with the contract. The adversary $\mathcal{A}$ is assumed to have full control over the communication channel between vehicles and the miners in the blockchain. This may include attacks like eavesdropping, tampering, replaying, and injecting packets in the network. The following set of queries is used to model these attacks:

- $\texttt{Send}S(S, \texttt{m0}, \texttt{r0}, \texttt{m1})$ is used to model the query where $\mathcal{A}$ acts like a legitimate vehicle and sends a message $\texttt{m0}$ to S and receives $\texttt{r0}$. The vehicle then replies to S with $\texttt{m1}$.
- $\texttt{Send}M(ID, \texttt{m0}, \texttt{r0})$ is used to model the query where $\mathcal{A}$ acts like a blockchain and sends a message $\texttt{m0}$ to a vehicle and receives $\texttt{r0}$.
- $\texttt{Monitor}(M,\ S)$ models $\mathcal{A}$'s ability to continuously eavesdrop on the channel between vehicle V and S.
- $\texttt{Drop}(\mathcal{A})$ models $\mathcal{A}$'s ability to drop packets between V and S.

The adversary $\mathcal{A}$ can call $\texttt{Send}S$, $\texttt{Send}M$, $\texttt{Monitor}$, and $\texttt{Drop}$ any polynomial number of times.

C. Security proofs

Lemma 1. *It is not possible for an adversary to tamper with the data in a blockchain.*

Proof. A blockchain is composed of chronological blocks hashed together, starting from the genesis block up until the latest block. Therefore, to tamper with the data in a single block, an adversary needs to successfully redo the PoW for that block and all the preceding blocks in the blockchain as well. However, the adversary needs to have at least 51% of the total computational power of the blockchain network to achieve this [33]. Given a decent sized blockchain network, such attacks are extremely difficult or even impossible. □

Lemma 2. *It is not possible to reveal the secret response of a vehicle.*

Proof. Every vehicle has its own PUF. During the authentication phase, a miner sends a challenge C^i to the vehicle and the vehicle uses this challenge to generate the secret response R^i. Thus, the vehicle does not store the secret response R^i in its memory and only generates it when needed. Therefore, even if an adversary launches a physical attack on an intelligent vehicle, he/she cannot obtain the secret response R^i. This shows that an adversary has no possible way of extracting/revealing the secret response for a vehicle. □

Theorem 1. *PUFs achieve mutual authentication of a vehicle and blockchain.*

Proof. An adversary $\mathcal{A}$ may try to authenticate itself as a legitimate vehicle. We can model this by the following game between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$:

1) $\mathcal{C}$ selects a legitimate vehicle V_1 and registers it with a miner in the blockchain.
2) $\mathcal{A}$ calls $\texttt{Send}S$, $\texttt{Send}M$, $\texttt{Monitor}$, and $\texttt{Drop}$ a polynomial number of times on the miner and vehicle V_1.
3) $\mathcal{A}$ invokes the $\texttt{Send}S$ oracle to authenticate itself as a legitimate vehicle to the miner.
4) $\mathcal{A}$ wins the game if he/she can successfully complete the authentication phase.

In the authentication phase, a vehicle needs to generate its secret response R^i to successfully authenticate itself with the miner, i.e., to pass the verification process, the vehicle needs to successfully create the authentication parameter $MAC(cID_V, ID_S, M_V, N_1, N_2, R^i)$. Therefore, when $\mathcal{A}$ attempts to authenticate itself with the miner, he/she also needs to produce a valid authentication parameter. However, to construct a valid authentication parameter, the adversary needs R^i. By lemma 2, this is not possible. Thus, an adversary cannot successfully authenticate itself as a legitimate vehicle.

The second part of the proof considers the case when an adversary $\mathcal{A}$ attempts to act as a miner and tries to fool a vehicle into authenticating it as the miner. This can be modeled by a security game similar to the one above except that in step 3, instead of calling the $\mathtt{Send}S$ oracle, $\mathcal{A}$ calls $\mathtt{Send}M$ to impersonate the legitimate miner. To successfully impersonate the miner, $\mathcal{A}$ needs to send a valid authentication parameter $MAC(cID_V, N_1, N_2, R^i)$ to the vehicle. However, by lemma 2, $\mathcal{A}$ cannot obtain R^i. Thus, we can conclude that successful authentication of both the miner and vehicle is achieved. □

Theorem 2. *Data provenance: The proposed protocol successfully establishes the authenticity of the origin of data.*

Proof. An adversary may try to impersonate a legitimate vehicle and send tampered data to a miner. We can model this by the following security game:

1) $\mathcal{C}$ selects a vehicle V_1 and uses it to perform a transaction.
2) $\mathcal{A}$ calls $\mathtt{Send}S$, $\mathtt{Send}M$, `Monitor`, and `Drop` a polynomial number of times on the miner and the vehicle.
3) $\mathcal{A}$ calls the $\mathtt{Send}S$ oracle to impersonate an intelligent vehicle.
4) If $\mathcal{A}$ can get the tampered data sent by it to the miner successfully valiated, then $\mathcal{A}$ wins the game.

To prove his/her legitimacy and successfully tamper with the data of the vehicles, $\mathcal{A}$ has two options: first, tamper with the blockchain and second, tamper with the hashed data parameter in message 2 during the authentication phase, i.e., $M_v = H(cID_V \parallel Data \parallel R^i)$. However, by lemma 1, tampering with the blockchain is not possible. Moreover, by lemma 2, the adversary cannot obtain R^i and thus cannot tamper with the hashed data parameter. This shows that data provenance is achieved and data tampering attacks are avoided. □

Lemma 3. *PUFs are safe against physical/cloning attacks.*

Proof. Physical attacks can be used by an adversary to extract secret keys from the memory of an intelligent vehicle. However, as shown in Lemma 2, vehicles do not store the secret response R^i (used to establish the various security properties) in their memory. Moreover, due to the SoC assumption, the PUF cannot be separated from a vehicle and neither can an adversary eavesdrop on the communication between the PUF and the vehicle's microcontroller. This provides a defense mechanism against physical attacks. □

Lemma 4. *The public keys of vehicles cannot be correlated.*

Proof. The RSUs in the proposed protocol issue certificates to randomize the public keys of vehicles. Thus, without access to an RSU, an adversary cannot correlate the public key of a vehicle for the current transaction with that of the next or previous one. □

Algorithm 2: Simulating the protocol

```
while $simulation$ do
    for $i\ in\ r_*,\ r_* \in R\ \forall * = 1, 2, 3, \cdots, r$ do
        $genesis.block \leftarrow define$
        $r_*[i] \leftarrow create\ node$
        $r_*[i].node \leftarrow make\ account$
        $r_*[i].node.account \leftarrow assign\ resource$
        for $j \leftarrow 1, i$ do
            $r_*^i[j].node \leftarrow compile\ contract$
            $r_*^i[j].node \leftarrow deploy$
    for $n\ in\ v_*,\ v_* \in V\ \forall * = 1, \cdots, v$ do
        $genesis.block \leftarrow define$
        $v_*[n] \leftarrow create\ node$
        $v_*[n].node \leftarrow make\ account$
        $v_*[n].node.account \leftarrow assign\ resource$
    while $r_*\ \&\ v_*$ do
        $contract \leftarrow sendMessage() \leftarrow v_*[n].node$
        $r_*^i[j].node \leftarrow contract \leftarrow v_*[n].node$
        if $request(v_*[n].node)$ then
            $Algorithm\ 1 \leftarrow call$
```

V. Implementation and Simulation

The enforcer contract was designed using Solidity (i.e., the programming language for writing smart contracts) to create the IoV-blockchain network model while the dPoW consensus mechanism was coded in Python (version 3.7.3). Note that the vehicles are assumed to be embedded with PUFs with their respective CRPs stored in the blockchain.

A. Setup

The simulations were carried out using a laptop with Ubuntu OS (version 17.04) that was installed on a virtual machine client, Oracle VM VirtualBox. The laptop had the following specifications: Intel core i7-7700HQ CPU (4 cores @2.8GHz), 16 GB RAM, Nvidia GeForce GTX 1060 GPU with 6GB memory, and 1 TB HDD with 128 GB SSD of storage. The shell scripting environment used was Terminal. The Ethereum development package was used in Terminal to instantiate the nodes using the Ethereum Go client (Geth), i.e., a command line interface written in Go language. Two types of nodes were initialized according to Algorithm 2, which includes a set of RSU nodes denoted by R and a set of vehicle nodes denoted by V. Each node has its own blockchain account and can interact with other nodes via the contract. Note that each RSU node holds a snapshot of the blockchain that is synced with its peer RSU nodes.

Furthermore, the Remix integrated development environment (IDE), a browser-based IDE for Solidity, was used for writing and compiling the smart contract. In addition, $web3.js$ (i.e., the official Ethereum JavaScript API) was used with the RSU and vehicle nodes. $web3.js$ is a collection of libraries that allow subject-object (RSU-vehicle) pair to interact with a local or remote Ethereum node using an HTTP/IPC connection.

At the vehicle side, $web3.js$ was used to interact with the corresponding Geth client via HTTP connections for sending requests to the contract as transactions and also to receive the outcome for its authorization. At the RSU side, $web3.js$ was used to interact with the Geth client for deploying the compiled contract and hosting the local blockchain.

TABLE III: Cost estimates of the smart contract

Function	Gas cost in Wei
$vehicleDelete(cID)$	42953
$vehicleRegister(cID)$	62898
$server()$	407
$sendMessage(string)$	∞
$vehicleCRPs()$	304
$registryCertificates()$	348
$registryVehicles()$	∞
$codeDepositCost$	985200
$executionCost$	21128

B. Execution flow of the contract

The enforcer contract needs to be compiled first before it can be deployed, i.e., the contract has two phases of operation: initialization and deployment. In the initialization phase, an RSU node initiates the contract which will be known as the $server$ variable by the contract instance. Thus, the contract can be compiled in the Geth terminal of RSU nodes. In the deployment phase, the compiled contract is deployed on the RSU nodes, which then proceed with mining to obtain the contract address required for interacting with the contract instance. For interaction, the contract ABIs will also be required that can be obtained from the Remix IDE. The contract address is then broadcast among the vehicle nodes to enable them to interact and communicate with the RSU nodes via the contract. Thus, the RSU nodes can now authorize vehicles, register or remove them, and allow them to send requests as transactions.

C. Operating cost of the protocol

The cost measure of performing a task in Ethereum is called "gas", i.e., for every operation executed in Ethereum (e.g., making a transaction or deploying a smart contract), there is a specified cost expressed in terms of gas. It is measured in Wei as: 1 Wei = 10^{-18} Ether. For instance, the units of gas consumed for deploying a contract represents the capital cost for performing this task. Thus, the more complex a task is, the more gas it requires to execute. The gas consumption estimates for the IoV-blockchain protocol and its functions are listed in Table III, which were obtained using the Remix IDE. The amount of gas required for executing the enforcer contract is 21128 while that for deploying it is 985200. Notice that functions $sendMessage()$ and $registryVehicles()$ have an infinite gas estimate. This is because their input size is not defined. The reason for this is twofold: (1) to allow $sendMessage()$ to send varying lengths of data (limited to 512 bytes); (2) by not defining an input length for $registryVehicles()$, an arbitrary number of vehicles can be added without restriction. Since this function is updated only after registering or removing a vehicle, it is safe to design it this way.

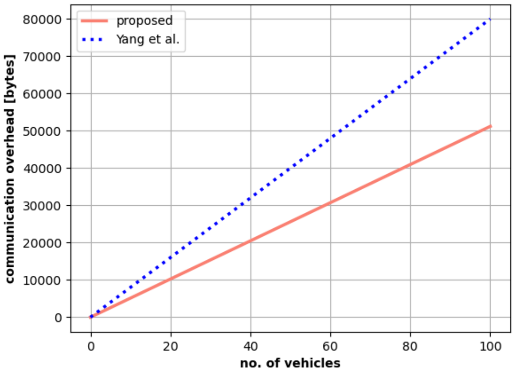


Fig. 4: Scalability of the proposed protocol.

VI. The Four-way Trade-off of IoV-blockchain - A Performance Analysis

This section presents a performance evaluation of the proposed protocol as well as discusses its efficiency in terms of the four-way trade-off of blockchain. We also compare the performance of the proposed protocol with [21] for driving trust management in IoV.

A. Scalability

By definition, a blockchain consists of a chain of blocks where each block contains both meta-data (previous hash) and data (tuples of transactions). This signifies the constantly increasing length of a blockchain with time. Therefore, we evaluate the scalability of the proposed protocol in dual terms: (i) communication overhead; (ii) dynamic proof-of-work consensus.

TABLE IV: Dynamic proof-of-work consensus thresholds

δ	Arrival rate	Difficulty level	PoW target consideration
1	low	high++	SHA256[0:4]
2	medium-low	high	SHA256[0:3]
3	medium-high	low	SHA256[0:2]
4	high	low++	SHA256[0:1]

1) Communication overhead: Let us consider the maximum packet size required for transmission by the proposed protocol with [21], as given in Table VII. We observe that the communication overhead for the proposed protocol is 36% lower than the protocol in [21]. To study the scalability of the proposed protocol in terms of communication overhead, we show the effect of increasing the number of vehicles on the communication overhead in Figure 4. It can be seen that the proposed protocol is more scalable than [21] and is able to manage more vehicles with less overhead.

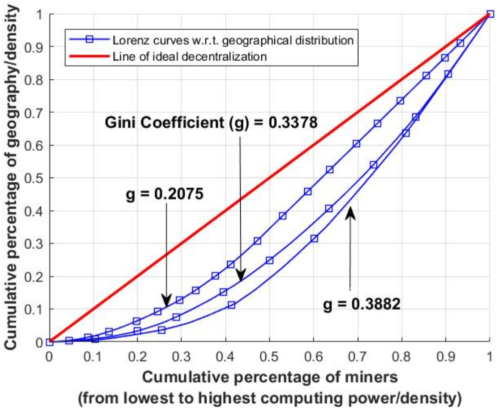


Fig. 5: Geographical decentralization of miners.

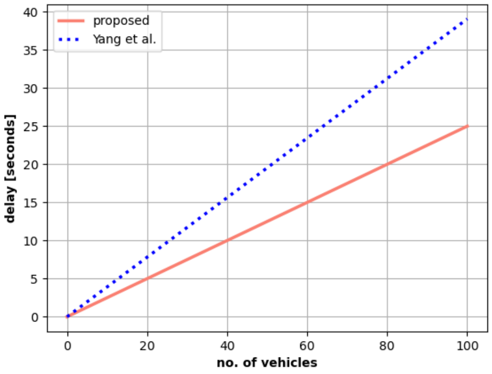


Fig. 6: Authentication delay at RSUs.

2) Dynamic proof-of-work: We define the dPow consensus algorithm with four scenarios listed in Table IV. We observe that when the arrival rate is low, mining difficulty is the highest with a mining target value defined by four significant bits of the mining hash. Note that SHA256 is considered as the hashing algorithm for hashing the blocks generated in the proposed protocol. Similarly, when the arrival rate is high, mining difficulty is the lowest with a target value defined by one significant bit of the mining hash. This allows the protocol to scale effectively, i.e., low arrival rate with high mining difficulty accounts for increased security fidelity while high arrival rate with low mining difficulty enables higher throughput, thereby mining more blocks in less time.

We simulated the scenarios mentioned in Table IV for 1000 blocks with 100 dummy transactions in each block, which are shown in Figure 7. We observe that the total time taken to mine the 1000 blocks under scenario $\delta = 4$ is almost 5 seconds (i.e., a throughput of 20,000 transactions per second $[tx/s]$), as can be seen in Figure 7(a). Figures 7(c) and (e) show that the total time required to mine under scenario $\delta = 3, 2$ are approximately 50 and 650 seconds (i.e., 2,000 tx/s and 150 tx/s), respectively. Moreover, it can be seen in Figure 7(g) that the total time consumed to mine the blocks under scenario $\delta = 1$ is 10,000 seconds (i.e., 10 tx/s). Thus, it can be concluded that for low arrival rates, the throughput is lower while for high arrival rates, the throughput is also higher, which makes the proposed protocol scalable.

B. Decentralization

The degree of decentralization for a blockchain can be evaluated by the distribution of its control and resources among its miners. To handle the requests generated as transactions in the IoV-blockchain network as well as ensure its smooth operation, the miners (RSUs) need to verify transactions and generate blocks efficiently. Therefore, the miners need to complete the following steps: (i) collect, verify, and collate the transactions into a block and mine it; (ii) broadcast the mined block in the network to reach a consensus and append it to the blockchain.

To measure the degree of decentralization for the proposed protocol, consider an IoV-blockchain system with N peer nodes and M miner nodes. Note that peer nodes represent both miner and server nodes. The set of nodes is denoted by $\mathcal{N} = \{n_1, n_2, \cdots, n_N\}$ and the computing power of node $n_n,\ n = 1, \cdots, N$, is represented by Υ_n. For clarity, the notation $\Upsilon = \{\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_n\}$ is used to represent the set of computational resources, i.e., computing power. Note that the M miners, represented by $\mathcal{M} = \{m_1, \cdots, m_m, \cdots, m_M\},\ \mathcal{M} \subseteq \mathcal{N}$, are selected out of $\mathcal{N}$. Assume that the miners are located at independent random positions in $\mathbb{R}^2$ according to an inhomogeneous Poisson point process (PPP) with density $\lambda(x)$ [40], where the location of node n_n is represented by the two-dimensional coordinates $x_n \in \mathbb{R}^2$ and $x = \{x_n\}$ is the location set. The density $\lambda(x)$ is defined such that $\mathbb{E}\{Num(A)\} = \int\int_A \lambda(x)dx$ for any $A \subseteq \mathbb{R}^2$, where $Num(A)$ is the number of nodes in area A. Note that the M miners take turns to mine and generate blocks with different mining difficulty levels.

To measure the decentralization degree, this paper uses the Gini coefficient, which is a well-studied measurement for inequality of wealth or income [41], [42]. We measure the decentralization of the propsoed protocol by considering the distribution of geographic locations of the RSUs. Since the miner density distribution $\lambda(x)$ is a continuous function of x, the Gini coefficient for miners with respect to (w.r.t.) geographical distribution can be expressed as [42]:

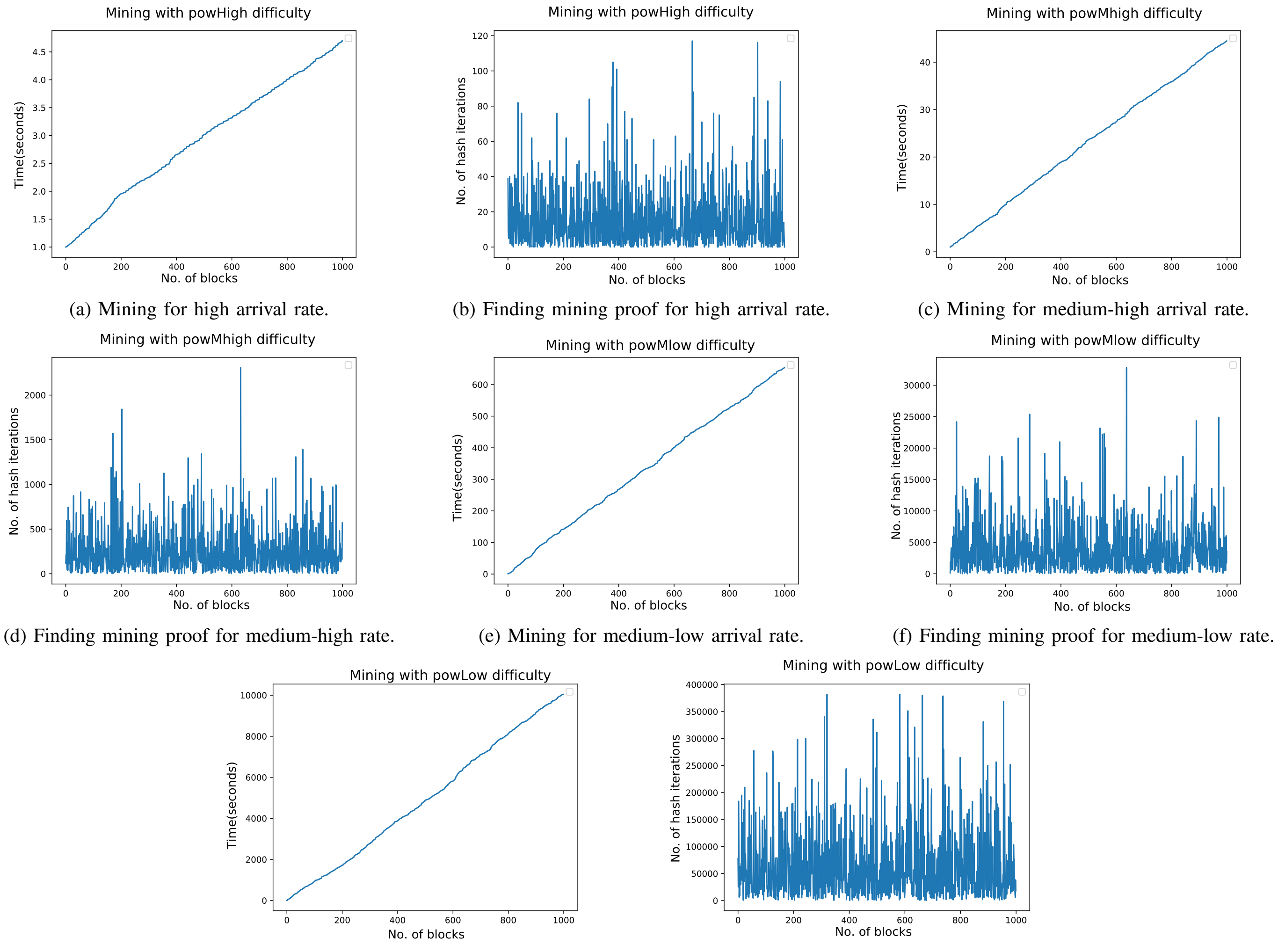
$$g(\lambda) = \frac{\displaystyle\int_a\int_a |\lambda(x) - \lambda(y)|dydx}{\displaystyle\int_a\int_a \lambda(x)dydx} = \frac{\displaystyle\int_a\int_a |\lambda(x) - \lambda(y)|dydx}{2M}, \tag{2}$$

where a represents the area relative to two dimensional coordinates (x, y) with the density set: $\lambda = \{\lambda(x)\},\ x \in a$, and the miners scattered in region $a \subseteq \mathbb{R}^2$.

Note that the values of Gini coefficient are within $[0, 1]$ range, where 0 denotes full decentralization and 1 denotes full centralization. Thus, more decentralized or uniform the geographical distribution of miners is, closer the coefficient is to 0. Moreover, to guarantee the geographical decentralization of miners, the following constraint needs to be satisfied:

$$g(\lambda) \leq \eta_g,\ \forall\ \eta_g \in [0, 1] \tag{3}$$

where η_g is the decentralization threshold w.r.t. geographical



(a) Mining for high arrival rate.

(b) Finding mining proof for high arrival rate.

(c) Mining for medium-high arrival rate.

(d) Finding mining proof for medium-high rate.

(e) Mining for medium-low arrival rate.

(f) Finding mining proof for medium-low rate.

(g) Mining for low arrival rate.

(h) Finding mining proof for low arrival rate.

Fig. 7: Mining comparison at RSUs with varying difficulty levels for different traffic arrival rates by vehicles.

distribution of the miners. Figure 5 presents the decentralization performance of the proposed protocol. It can be seen that as the value of Gini coefficient decreases, the Lorenz curve gradually approaches the line of ideal decentralization, i.e., the blockchain becomes more decentralized. From the figure, different areas of Lorenz curve can be seen for three different values of Gini coefficient. Note that a higher value of Gini coefficient means that the miners are distributed more centrally while a lower value means they are distributed in a decentralized manner.

C. Latency

Similar to MAC/PHY overhead, latency is also an important performance metric for IoVs as it signifies the total time required for a vehicle to successfully authenticate itself and its request thereafter, with an RSU. Therefore, the latency of the proposed protocol is evaluated in dual terms: (i) we study the authentication delay at the RSU side, i.e., the time required for an RSU to authenticate a vehicle; (ii) we study TiF (cf. Section I-A) for transactions (requests) generated by the vehicles, i.e., the time taken to receive a reasonable guarantee that a transaction has been written in the blockchain, or in other words, is finalized and irreversible.

1) Authentication delay at RSU: Figure 6 shows the authentication delay of vehicles at the RSU side for the proposed protocol as well as the protocol in [21]. For a fair comparison, we simulated both protocols by using the parameters given in Table VI. The results were obtained using a custom discrete-event simulator in MATLAB, which confirm that the authentication delay of the proposed protocol is significantly lower than [21]. This reduces the load on RSUs and enables them to effectively manage trust as the number of vehicles increases.

2) Time-to-finality (TiF): For TiF, we consider the same four scenarios that are listed in Table IV. Transaction processing in the proposed protocol requires two steps: adding transactions together to form a block and mine it, and then reach a consensus on the mined block. Thus, we formulate TiF for transactions generated by a vehicle, which includes the block mining time (i.e., block interval) and the block verification time as:

$$t_{f,\delta} = t_{i,\delta} + t_{c,\delta}, \quad \delta = 1, 2, 3, 4 \tag{4}$$

where $t_{i,\delta}$ represents the block interval time that depends on the choice of δ while $t_{c,\delta}$ represents the consensus latency, i.e., the time that miners need to verify a block. For simplicity, we assume that the consensus latency time is one, since verifying a block is easy and involves inputting the proof value to generate the target hash. Therefore, we only consider the block interval time and primarily focus on the mining dynamics of the protocol under different arrival rates of transactions.

We simulated the scenarios in Table IV for 1000 blocks with 100 dummy transactions each, which are shown in Figure 7. The number of hash iterations for each block (i.e., TiF for the 100 transactions in it) under scenario $\delta = 4$ is shown in Figure 7(b) with a peak value of 120 hash iterations per block (h/b). Similarly, the number of hash iterations under scenario $\delta = 3, 2$ are shown in Figures 7(d) and (f) with peak values of 2500h/b and 35000h/b, respectively. Moreover, Figure 7(h) shows the number of hash iterations for each block under scenario $\delta = 1$ with a peak value of approximately 400,000h/b. Thus, we can conclude that for low arrival rates, TiF is higher while for high arrival rates, TiF is lower, which enables the proposed protocol to operate with low latency.

Thus, vehicles are expected to receive the finality of transactions within short periods of time in the proposed protocol. To meet the IoV-blockchain delay requirement, it is assumed that one block should be mined and verified within a number of consecutive block intervals, i.e., Δ ($\Delta > 1$) block intervals. Specifically, the TiF should satisfy the following constraint:

$$t_{f,\delta} \leq \Delta t_i, \quad \delta = 1, 2, 3, 4. \tag{5}$$

It can also be observed that the effect of block interval on a blockchain is twofold: (i) reducing block interval can improve throughput, as depicted in Figure 7 and (ii) TiF increases proportionally with larger block intervals since large intervals translate into more transactions that require verification. Moreover, the decrease of block interval imposes a stricter constraint on the consensus delay, which is closely related to miners and the dynamics of the chosen consensus algorithm. Therefore, the adjustment of block interval, distribution of miners, and selection of consensus dynamics should be conducted carefully for addressing the four-way trade-off.

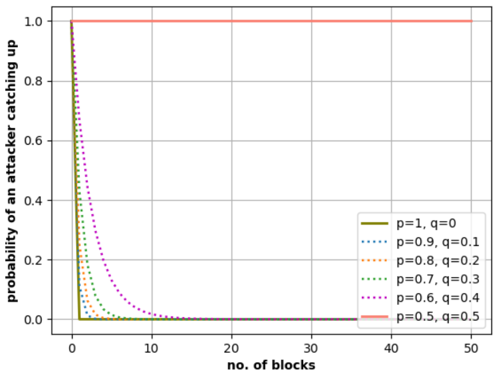


Fig. 8: Probability of an attacker reaching a break even point in an honest chain (at least 51% honest miners).

D. Security

The proposed protocol uses a variant of the PoW consensus algorithm, i.e., dPoW. PoW offers a high degree of security as long as the number of honest miners is greater than the malicious ones. In theory, an adversary/group of adversaries can potentially mine an alternative "longer" chain that goes back to the genesis block by using (>50%) mining power [43]. Therefore, the loyalty of honest miners is very critical in PoW. To guarantee the security of the proposed protocol with consensus algorithm δ, the number of malicious miners n_m needs to be restricted by the following constraint:

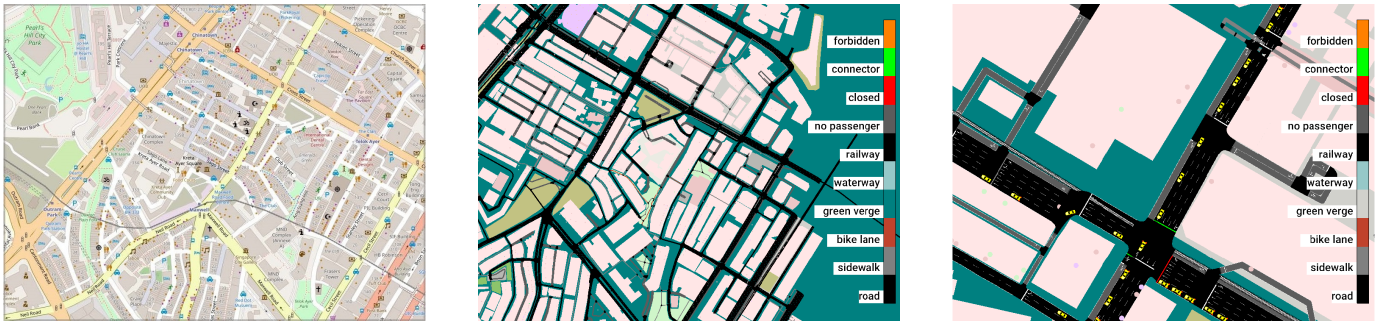
$$n_m \leq n_{m,\delta}, \;\; \delta = dPoW \tag{6}$$

where $n_{m,\delta} = \lfloor \frac{n_{h,\delta}-1}{2} \rfloor$ denotes the maximum tolerable number of malicious miners in the proposed protocol, i.e., <50%, and $n_{h,\delta}$ represents the total number of honest miners. To analyse this, we consider a scenario where an attacker is trying to create an alternate (dishonest) chain faster than the honest miners' one (i.e., honest chain). The competition between the attacker and honest miners can then be characterized as a binomial random walk. Here, success represents the honest chain being extended by one block, thereby increasing its lead from the dishonest chain by $+1$. On the other hand, failure represents the dishonest chain being extended by one block, thereby reducing the gap by -1. Moreover, the probability of the attacker catching up from n blocks behind the honest chain is analogous to a Gambler's Ruin problem. Suppose a gambler starts at a given deficit with unlimited credit and potentially plays an infinite number of trials to try to reach the break even point. The probability that the attacker catches up with the honest chain can is then given by [44]:

$$Q_n = \left\{ \begin{matrix} 1 & if \; p \leq q \\ (\frac{q}{p})^n & if \; p > q \end{matrix} \right\}, \tag{7}$$

where p is the probability that an honest miner finds the next block, q is the probability that the attacker finds the next block, and Q_n is the probability that the attacker will catch up from n blocks behind the honest chain. Figure 8 demonstrates the ineffectiveness of this attack as long as the honest miners have more than 50% of the total computational capacity. It can be seen that when $p = 1, 0.9, 0.8, 0.7,$ and 0.6, the probability of the attacker Q_n catching up with the honest chain decreases exponentially with the increasing number of blocks, i.e., after just 10 blocks, Q_n reduces to 0. When $p = 0.5$, Q_n increases to 1, which confirms that whoever controls more than 50% of the total computational capacity of the proposed protocol, controls its blockchain. However, given our assumption $p > q$, Q_n drops exponentially as the number of blocks the attacker has to catch up with increases, thereby invalidating the attack.

Moreover, authenticity of transactions is guaranteed via digital signatures. The proposed protocol uses Elliptic Curve Cryptography (ECC) with Elliptic Curve Digital Signature Algorithm (ECDSA) to ensure that data generated by vehicles is legitimate. A common comparison between ECC and Rivest Shamir Adelman (RSA), Diffie-Hellman (DH), and Digital Signature Algorithm (DSA) is given in Table V [45]. We observe that ECC can achieve higher levels of security using



(a) The Chinatown region of Singapore.

(b) The extracted map segment.

(c) Traffic in the map segment (zoomed).

Fig. 9: Simulating a real-life traffic scenario.

TABLE V: Security strength comparison of key size combinations for various cryptographic algorithms

Key size (bits)				Ratio
Security	Symmetric encryption algorithm	ECC	RSA/DH/DSA	
80	Skipjack	160-223	1024	1:6-30
112	3DES	224-255	2048	
128	AES-128	256-383	3072	
192	AES-192	384-511	7680	
256	AES-256	512-more	15360	

smaller key lengths and can provide the same security level afforded by a RSA-based system with a large modulus, and correspondingly larger key. Note that smaller key lengths translate into lower computational overhead [46].

VII. Case Study

We designed a case study for our protocol using the Urban Mobility (SUMO[1]) simulation package with OSMWebWizard[2]. SUMO is an open source, microscopic, highly portable, and continuous road traffic simulation package designed to handle large road networks. We considered the region of Chinatown in Singapore with 100 left-hand driving nodes that include vehicles, buses, and trucks, moving at a uniform speed of 33 meters/second with no pause time. Figure 9 shows the steps taken to extract the considered map segment, which has a route length of approximately 1300 meters. After extracting the map segment, a map configure (.cfg) file was generated by SUMO which was converted to a vehicle trace file (.xml) first, and then a vehicle mobility trace file (.tcl) was generated using the 'traceExporter' function offered in the SUMO package.

A. *The IoV-blockchain protocol dynamics*

To analyze our protocol, we used ns-3[3], which is a discrete-event network simulator for Internet systems. With the mobility trace file, we defined a custom VANET using the wireless access for vehicular environments (WAVE) protocol and the parameters listed in Table VI. The standard for the WAVE protocol was IEEE 802.11p @5.9GHz (current state-of-the-art for vehicular environments) with continuous access to a 10 MHz control channel for all traffic generated by the vehicles.

TABLE VI: Key parameters

Parameter	**Value**
WAVE ITS band	5.9GHz
routing protocol	OLSR
no. of RSUs	10
inter-RSU distance	50:500m
transmitting power	20dBm
propagation loss model	Two-ray ground
fading model	Nakagami fading
physical mode	OFDM @6Mbps
channel bandwidth	10MHz
packet interval	100ms
transmission max. delay	10ms
antenna height	1.5m
physical layer radio	DSSS @11Mbps
transmission rate	2.048Kbps
blockchain packet size	512-byte
application packet size	64-byte
simulation time	100s

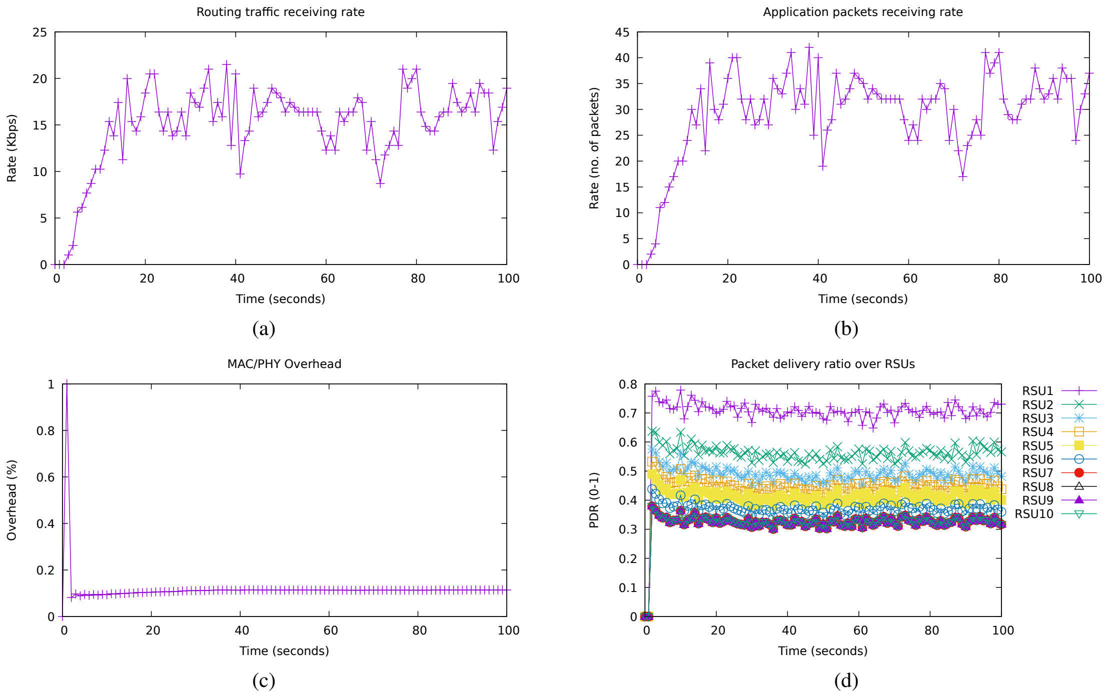
We considered both blockchain packets (i.e., block headers) as well as application packets using the OLSR routing protocol. The simulation was run for 100 seconds with 100 vehicles and 10 RSU nodes, respectively. Moreover, the vehicle nodes were moving according to the mobility trace file within the 650×650 m^2 region of the extracted map segment.

All vehicle nodes transmit a 512-byte block header 10 times/second to the RSUs at a rate of 6 Mbps, i.e., 10,000 block headers being broadcast per unit time (second). A block header in the proposed protocol is approximately 508 bytes [35]. However, to account for losses on runtime, we considered 512 bytes. Note that RSUs act as sink nodes as well as miners. Additionally, all vehicles attempt to continuously send 64-byte application packets at a rate of 2.048 Kbps to other nodes. The routing protocol used by the vehicles was OLSR and the two-ray ground loss model was used in conjunction with the Nakagami fading model. The transmitting power of vehicles was set to 20 dBm and the RSUs are placed 50 meters apart. We calculated the delivery of blockchain packets in terms of packet delivery ratio (PDR) which is the number of packets sent to an RSU divided by the number of packets it actually

[1] https://sumo.dlr.de/docs/index.html

[2] https://sumo.dlr.de/docs/Tutorials/OSMWebWizard.html

[3] https://www.nsnam.org/



(a)

(b)

(c)

(d)

Fig. 10: The different dynamics of the IoV-blockchain protocol.

receives. We did this for all the 10 RSUs to see the effect of fading over distance.

B. MAC/PHY overhead

The MAC and physical layers (MAC/PHY) overhead is an important factor in IoVs. Typically, in existing trust management protocols for vehicular networks, the broadcast channel is flooded with authentication requests of the vehicles. This affects the application throughput and may also overload the RSUs, resulting in longer authentication delays. Therefore, MAC/PHY overhead is critical for the performance of IoVs. Its value lies in the range $[0, 1]$ and is given by:

$$MAC/PHY overhead = (totalPHYBytes - totalAppBytes)/totalPHYBytes, \quad (8)$$

where PHYBytes represent traffic at Layer 1 (physical) while AppBytes represent traffic at Layer 2 (data link). The MAC/PHY overhead can be seen as a moving average of the overhead of a protocol over time.

C. Discussion and evaluation

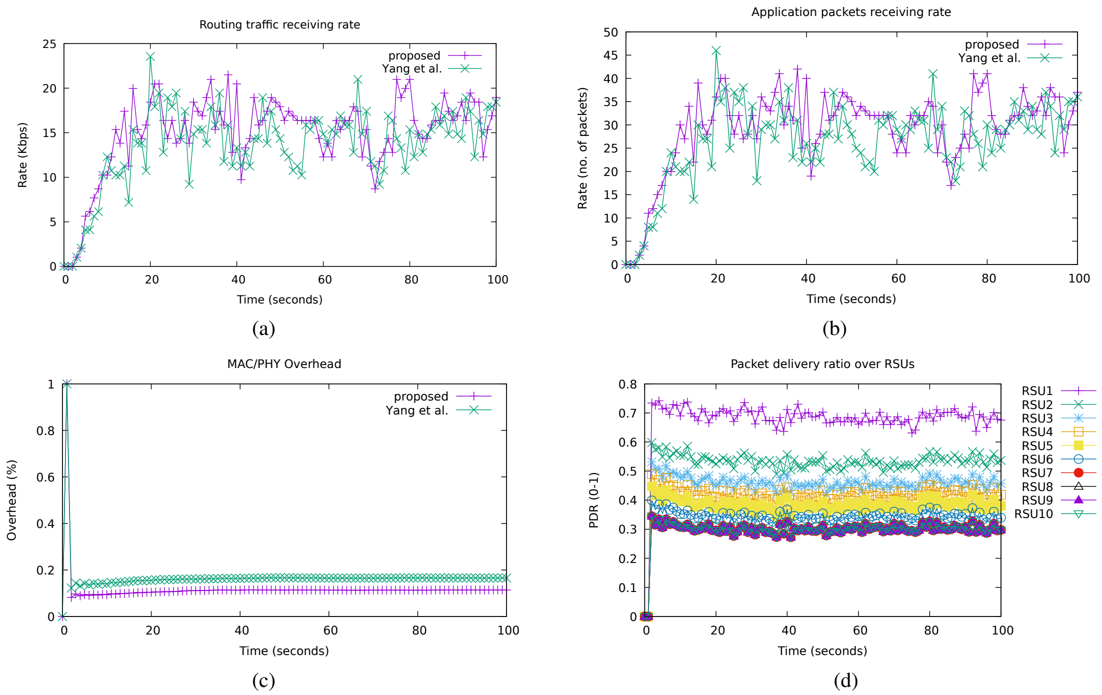
Figure 10(a,b) show the application data and application packet receiving rates of the protocol. Note that the number of bits/second transmitted represents data rate and recall that the vehicle nodes are transmitting 64-byte application packets at a rate of 2.048 Kbps. We can see that the total application throughput remains at an average of 150 Kbps while the application packet receiving rate remains at an average of 30 packets per second (pps). We observe that our protocol achieves the upper bound for throughput in IEEE 802.11p in the current scenario [47]. Figure 10(c) shows the associated MAC/PHY overhead whose low value demonstrates the efficiency of the proposed protocol. Moreover, Figure 10(d) shows the PDR for the 10 RSUs, which represents the blockchain packets broadcast to the RSUs by the vehicles. Being the nearest, RSU1 has the highest PDR while RSU10, being the furthest, has the lowest due to channel fading. Thus, we can conclude that these results show the effectiveness of the proposed protocol in IoV.

D. A comparative analysis

In this section, we compare our protocol with the state-of-the-art protocol proposed by Yang et al. [21]. For a fair comparison and without loss of generality, we simulated both protocols using the parameters explained in Section VII-A. The simulation results and improvements are documented in Table VII while Figure 11 presents a graphical depiction of the protocol dynamics comparison. We observe that the proposed protocol outperforms [21] in all of the comparison metrics.

Figure 11(a) shows that the total receiving rate of application traffic for the protocol in [21] remains at an average of 140 Kbps as compared to 150 Kbps of the proposed protocol, i.e., a 7.15% improvement in the receiving rate. Figure 11(b) shows that the application packet throughput for [21] is around 25 pps as compared to 30 pps of the proposed protocol. This is because the transmitted packet in [21] has a size of 800 bytes while in the proposed protocol, it is 512 bytes.

Similarly, Figure 11(c) shows that the MAC/PHY overhead of the proposed protocol is lower than the protocol in [21]. We observe that after 100 seconds, the MAC/PHY overhead of [21] is at 16.55% while that of the proposed protocol is 11.39%. Moreover, comparing Figures 10(d) and 11(d), it can be seen that the proposed protocol also outperforms the



(a)

(b)

(c)

(d)

Fig. 11: Comparative analysis of the IoV-blockchain protocols.

protocol in [21] in terms of PDR at different RSU locations. Thus, due to a higher throughput with lower overhead and packet size, the proposed protocol is able to drive trust management for more vehicles in lesser time, which can be verified by the RSU PDRs detailed in Table VII. This also results in good scalability of the proposed protocol.

TABLE VII: Comparison of key parameters

Parameter	Proposed protocol	Yang et al. [21]	Improvement (%)
Transmitted packet size (bytes)	512	800	36
Throughput (pps)	30	25	20
MacPhyOverhead reduction rate (%ps)	0.886069	0.834459	5.83
PDR (RSU1)	0.70873	0.688423	2.8
PDR (RSU2)	0.563481	0.534726	5
PDR (RSU3)	0.490924	0.461358	6
PDR (RSU4)	0.449408	0.420628	6.4
PDR (RSU5)	0.413202	0.384506	6.9
PDR (RSU6)	0.37116	0.344836	7
PDR (RSU7)	0.327475	0.302728	7.5
PDR (RSU8)	0.327475	0.302728	7.5
PDR (RSU9)	0.327475	0.302728	7.5
PDR (RSU10)	0.327475	0.302728	7.5

VIII. Conclusion

This paper investigated the issue of providing trust management in IoV and presented a blockchain based protocol, which uses smart contracts with PUF, certificates, and a dPoW consensus algorithm. The blockchain and contracts form the basis for a decentralized IoV network by managing vehicle registrations. PUF gives each vehicle a unique fingerprint via which trust is established. Certificates are issued by RSU which preserve the privacy of vehicles. Moreover, the dPoW consensus allows the protocol to scale according to the incoming traffic generated by the vehicles. The proposed protocol is also able to provide distinction between registered and malicious vehicles by managing a list of registered vehicles.

References

[1] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, Firstquarter 2016.

[2] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2018.

[3] U. Javaid, M. N. Aman, and B. Sikdar, "Drivman: Driving trust management and data sharing in vanets with blockchain and smart contracts," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–5.

[4] N. Malik, P. Nanda, A. Arora, X. He, and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *2018 17th IEEE Int. Conf. On Trust, Security And Privacy In Computing And Commun./12th IEEE Int. Conf. On Big Data Science And Eng. (TrustCom/BigDataSE)*, Aug 2018, pp. 674–679.

[5] M. N. Aman, K. C. Chua, and B. Sikdar, "Physically secure mutual authentication for iot," in *2017 IEEE Conference on Dependable and Secure Computing*, 2017, pp. 310–317.

[6] T. Gazdar, A. Rachedi, A. Benslimane, and A. Belghith, "A distributed advanced analytical trust model for vanets," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 201–206.

[7] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3661–3669, 2019.

[8] E. A. Brewer, "Towards robust distributed systems (abstract)," in *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '00. New York, NY, USA: ACM, 2000, pp. 7–. [Online]. Available: http://doi.acm.org.libproxy1.nus.edu.sg/10.1145/343477.343502

[9] K. Zhang and H. Jacobsen, "Towards dependable, scalable, and pervasive distributed ledgers with blockchains," in *2018 IEEE 38th Int. Conf. on Distributed Computing Systems (ICDCS)*, July 2018, pp. 1337–1346.

[10] M. Snider, K. Samani, and T. Jain, "Delegated proof of stake: features & tradeoffs," https://multicoin.capital/wpcontent/uploads/2018/03/DPoS-Features-and-Tradeoffs.pdf, March 2018, accessed: 17-08-2019.

[11] K. Samani, "Models for scaling trustless computation," https://multicoin.capital/2018/02/23/models-scaling-trustless-computation, accessed: 12-08-2019.

[12] M. N. Aman, B. Sikdar, K. C. Chua, and A. Ali, "Low power data integrity in iot systems," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3102–3113, 2018.

[13] U. Klarman, S. Basu, A. Kuzmanovic, and E. G. Sirer, “bloxroute: A scalable trustless blockchain distribution network whitepaper v1.0,” *Bloxroute Labs, Whitepaper*, March 2018.

[14] M. N. Aman and B. Sikdar, “Att-auth: A hybrid protocol for industrial iot attestation with authentication,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5119–5131, 2018.

[15] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, “Mitigating lot device based ddos attacks using blockchain,” in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, ser. CryBlock’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 71–76. [Online]. Available: https://doi.org/10.1145/3211933.3211946

[16] M. N. Aman, M. H. Basheer, S. Dash, J. W. Wong, J. Xu, H. W. Lim, and B. Sikdar, “Hatt: Hybrid remote attestation for the internet of things with high availability,” *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[17] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, “A job market signaling scheme for incentive and trust management in vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3657–3674, 2015.

[18] Y. Yahiatene and A. Rachedi, “Towards a blockchain and software-defined vehicular networks approaches to secure vehicular social network,” in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–7.

[19] Z. Lu, Q. Wang, G. Qu, and Z. Liu, “Bars: A blockchain-based anonymous reputation system for trust management in vanets,” in *2018 17th IEEE Int. Conf. On Trust, Security And Privacy In Computing And Communications/ 12th IEEE Int. Conf. On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug 2018, pp. 98–103.

[20] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, “A privacy-preserving trust model based on blockchain for vanets,” *IEEE Access*, vol. 6, pp. 45 655–45 664, 2018.

[21] 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, April 2019.

[22] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, “Blockchain-based dynamic key management for heterogeneous intelligent transportation systems,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, Dec 2017.

[23] H. Khelifi, S. Luo, B. Nour, H. Moungla, and S. H. Ahmed, “Reputation-based blockchain for secure ndn caching in vehicular networks,” in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2018, pp. 1–6.

[24] X. Zhang and X. Chen, “Data security sharing and storage based on a consortium blockchain in a vehicular adhoc network,” *IEEE Access*, pp. 1–1, 2019.

[25] X. Zhang, R. Li, and B. Cui, “A security architecture of vanet based on blockchain and mobile edge computing,” in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Aug 2018, pp. 258–259.

[26] M. Singh and S. Kim, “Trust bit: Reward-based intelligent vehicle commination using blockchain paper,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb 2018, pp. 62–67.

[27] R. Shrestha, R. Bajracharya, and S. Y. Nam, “Blockchain-based message dissemination in vanet,” in *2018 IEEE 3rd Int. Conference on Computing, Communication and Security (ICCCS)*, Oct 2018, pp. 161–166.

[28] M. Singh and S. Kim, “Crypto trust point (ctp) for secure data sharing among intelligent vehicles,” in *2018 Int. Conference on Electronics, Information, and Communication (ICEIC)*, Jan 2018, pp. 1–4.

[29] T. Jiang, H. Fang, and H. Wang, “Blockchain-based internet of vehicles: Distributed network architecture and performance analysis,” *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[30] A. Kaci and A. Rachedi, “Poolcoin: Toward a distributed trust model for miners’ reputation management in blockchain,” in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[31] M. N. Aman, K. C. Chua, and B. Sikdar, “A light-weight mutual authentication protocol for iot systems,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[32] U. Javaid, M. N. Aman, and B. Sikdar, “Blockpro: Blockchain based data provenance and integrity for secure iot environments,” in *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems*, ser. BlockSys’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 13–18. [Online]. Available: https://doi.org/10.1145/3282278.3282281

[33] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 03 2009.

[34] V. Buterin, “Ethereum: A next-generation smart contract and decentralized application platform,” https://github.com/ethereum/wiki/wiki/White-Paper, 2014, accessed: 2016-08-22. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[35] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[36] M. N. Aman, M. H. Basheer, and B. Sikdar, “Two-factor authentication for iot with location information,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3335–3351, 2019.

[37] S. Guilley and R. Pacalet, “Socs security: a war against side-channels,” *Annals of Telecommun.*, no. 59(7), pp. 998–1009, 2004.

[38] M. Kirkpatrick, S. Kerr, and E. Bertino, “System on chip and method for cryptography using a physically unclonable function,” Patent US 8 750 502 B2, issued March 22, 2012.

[39] C. Bohm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. Springer, 2012.

[40] D. Stoyan, W. Kendall, and J. Mecke, “Stochastic geometry and its applications,” *New York, NY, USA: Wiley*, vol. 76, p. 619–622, May 1996.

[41] C. Gini, “Variability and mutability,” *Journal of The Royal Statistical Society*, vol. 76, pp. 619—-622, May 1913.

[42] Z. Lin, F. Wen, Y. Ding, and Y. Xue, “Data-driven coherency identification for generators based on spectral clustering,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1275–1285, March 2018.

[43] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “When mobile blockchain meets edge computing: Challenges and applications,” *CoRR*, vol. abs/1711.05938, 2017. [Online]. Available: http://arxiv.org/abs/1711.05938

[44] W. Feller, *An Introduction to Probability Theory and Its Applications*. Wiley, January 1991.

[45] N. Z. Aitzhan and D. Svetinovic, “Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams,” *IEEE Trans. on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852, Sept 2018.

[46] M. N. Aman, S. Taneja, B. Sikdar, K. C. Chua, and M. Alioto, “Token-based security for the internet of things with dynamic energy-quality tradeoff,” *IEEE IoT J.*, pp. 1–1, 2018.

[47] Y. Wang, X. Duan, D. Tian, G. Lu, and H. Yu, “Throughput and delay limits of 802.11p and its influence on highway capacity,” *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 2096 – 2104, 2013, intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813023628

Uzair Javaid (S’19) received the B.Sc. degree in Electrical Engineering from FAST-National University of Computer and Emerging Sciences, Peshawar, Pakistan, where he graduated with Magna Cum Laude. He is currently a Ph.D. research scholar with the Department of Electrical and Computer Engineering at the National University of Singapore, Singapore. His research interests include blockchain, applied cryptography, and network security.

Muhammad Naveed Aman (S’12-M’17) received the B.Sc. degree in Computer Systems Engineering from KPK UET, Peshawar, Pakistan, M.Sc. degree in Computer Engineering from the Center for Advanced Studies in Engineering, Islamabad, Pakistan, M.Engg. degree in Industrial and Management Engineering and Ph.D. in Electrical Engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA in 2006, 2008, and 2012 respectively.

He is currently working as a Senior Research Fellow with the Department of Computer Science at the National University of Singapore, Singapore. Dr. Aman previously served on the faculty of National University of Computer and Emerging Sciences Pakistan as an Assistant Professor. His research interests include IoT and network security, wireless and mobile networks, and secure embedded systems.

Biplab Sikdar (S'98-M'02-SM'09) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University,Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was on the faculty of Rensselaer Polytechnic Institute from 2001 to 2013, first as an Assistant and then as an Associate Professor.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include wireless network, and security for IoT and cyber physical systems. Dr. Sikdar is a member of Eta Kappa Nu and Tau Beta Pi. He served as an Associate Editor for the *IEEE Transactions on Communications* from 2007 to 2012. He currently serves as an Associate Editor for the *IEEE Transactions on Mobile Computing*.