2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Secure Ramp Merging using Blockchain

Ahmed Abdo[1], Guoyuan Wu[2], Nael Abu-Ghazaleh[1]

Abstract— **Connected vehicles nowadays can provide a variety of useful and advanced services to their owners, manufacturers, transportation authorities, and other mobility service providers. Securing the complex sensing and networking protocols that enable these applications is an important and difficult problem. In this paper, we use *blockchain* which is traditionally used in applications from cryptocurrencies to smart contracts, as a potential solution to CV security. Specifically, we exploit the immutability of blockchain to ensure safety from falsified information and attacks. We demonstrate these properties by developing an algorithm that uses blockchain to maintain trusted communications between vehicles in the context of a cooperative ramp merging application.**

I. Introduction

Connected vehicles (CVs) can communicate with each other i.e., Vehicle-to-Vehicle or V2V, with infrastructure roadside units to control traffic signals or avoid accidents i.e., Vehicle-to-Infrastructure or V2I , or with other mobile devices that can be carried by certain passengers. Connected vehicle applications can vary from infotainment, parking assistance, roadside assistance, and remote diagnostics to supporting essential services for self-driving vehicles, in order to improve traffic safety, efficiency and sustainability [1][2]. Connected vehicles rely on Dedicated Short-Range Communication (DSRC) that offers highly stable connectivity, secure communication, and low latency for time-critical V2V and V2I applications. Other wireless communication prototypes have been explored such as mixtures of DSRC with WiFi, LTE and WiMAX communication technologies to include a reliable next-generation communication infrastructure for connected vehicles [3].

Security, privacy, and trust establishment can be a necessity for vehicular cooperation and communication networks[4],[5],[6]. Blockchain (BC), which in essence is a distributed ledger technology, has promising properties that help with the challenges facing CVs; these properties include decentralization, availability, transparency, immutability, and pseudonymity. Moreover, BC can provide a secure, scalable and privacy-preserving environment without relying on a trusted central authority. It is a promising technology for designing secure and trusted services or applications in vehicular networks at low cost. Blockchain[7] offers a new public infrastructure for verifying credentials in a secure manner, which is more convenient than relying upon only a single authority.

[2]Ahmed Abdo and Nael Abu-Ghazaleh are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA. aabdo003@ucr.edu, nael@cs.ucr.edu.

[1]Guoyuan Wu is with the Center for Environmental Research and Technology, University of California, Riverside, CA 92521, USA. gywu@cert.ucr.edu .

CV security is very critical, since a single malicious attacker may compromise safety for a large number of vehicles and endanger protection or disrupt the entire traffic flow. Therefore, ensuring that all CVs have timely and proper protections against spoofing attacks is important and challenging [8]. The variety of CV receivers, including infrastructure devices, vehicles, and pedestrians, in a CV environment increases to the system's sophistication. For example, security testers [9] showed how they could obtain unauthorized access to monitor or control the steering wheel, fans, seats, and air conditioning remotely for the Nissan Leaf vehicle. Thus, to profoundly solve this security dilemma, data spoofing must be prevented in a timely manner.

Having only digital signatures based on public key infrastructure (e.g., Security Credential Management System, the standard solution being deployed for CVs[10]) is important for a secure credentialing solution. However, verifying digital records through a trusted third party, to transmit or provide verification, is critical to this scheme. If the trusted third party is compromised, loses its records, or stops functioning, verification is no longer possible. For example[7], in 2017, Hurricane Maria hit Puerto Rico. Critical infrastructure was wiped out by the hurricane, causing loss of high-stakes records. These included vital records (birth, death, and marriage certificates), driver's licenses, property titles, and address and tax records. Entrusting a single entity with the power to protect and verify all the records creates a brittle system with poor security and longevity. It is insufficient for high-stake records that need to be accessed and verified reliably for a lifetime. A better alternative is to have this same trusted authority be decentralized: backed up numerous times across the system, and accepted across jurisdictions, because the data would not controlled by any single company or government organization. This decentralization is the main advantage of BC.

Therefore, in this paper we design a new scheme considering utilizing blockchain consensus mechanisms for CV security against cyber spoofing attacks. At its core, our system leverages BC to create a decentralized verification authority, with equivalent functionality to Security Credential Management System). In particular, we use a data-driven methodology to maintain trusted communications between vehicles in the context of a cooperative ramp merging application, supported using BC. Our results show that we are able to detect malicious vehicles in a quick manner (less than two seconds) using a BC implementation with low computational cost. We believe that our results demonstrate the feasibility and effectiveness of using BC to track trust in a CV environment.

2021 IEEE Intelligent Vehicles Symposium (IV) | 978-1-7281-5394-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IV48863.2021.9575411

Our key contributions in this study are:

- We develop a prototype of V2X security mitigation scheme based on BC technology and a data driven solution for malicious trajectory information.
- We show how BC consensus can be designed to avoid various spoofing attacks with the help of different vehicular units that act together to validate information coming from any suspicious actor.
- We perform extensive simulation to show efficiency and effectiveness of our scheme using an open source simulation framework.

The remainder of this paper is organized as follows: Section II conducts a literature review of two major topics: BC technology and BC applications in connected vehicles. Section IV introduces the design of the proposed scheme with detailed description of its components and associated algorithms. Results of our simulations are presented in Section VI. Lastly, Section VII presents some concluding remarks and outlines possible future work.

II. Background

Blockchain (BC) technology is a distributed ledger technology, enabling participants of the system to agree on a transaction and log it in an unforgeable shared ledger that can be used as a record of the agreement. We propose to leverage BC to support managing and maintaining historical transactions in a Connected Vehicle (CV) environment. This allows any node in the system (i.e., vehicle or roadside infrastructure) to access past event list and its related information in the blockchain, and use that for example, to establish trust in vehicles based on past behavior. In our scheme, we use BC to ensure data immutability and automated information exchanges between different trusted nodes in a safe manner. Moreover, we rely on a credit based consensus protocol which can be seen as a credit score system to estimate the trust level in a vehicle: the higher the node's credit score is, the higher the trust level of this node would be. Using these credit scores, it is possible to separate trusted from untrusted nodes, and take that information into account in critical maneuvers in the system. This ability is important for cooperative CV applications such as ramp merging [11] and intersection management [12], since they are time critical maneuvers for road safety and traffic efficiency where a malicious participant can substantially interfere with the system.

A. Consensus Protocols

BC [13] is a distributed ledger spreading across nodes which can be used to verify transactions on a P2P network. This is the key feature of BC that enables its unique decentralized property. It is important for BC to ensure agreement on which information is added or discarded. These processes or rules, are essentially known as a *consensus protocols* in the distributed computing community which ensure that a group of participants can reach consensus on a value even in the presence of malicious participants. Consensus is used to verify transactions and help keep the network safe.

A consensus protocol needs to be set up before the blockchain is created and it is the heart of a BC network. It provides a method of reviewing and confirming what data should be added to the blockchain's record. Because a BC network typically has no centralized authority to oversee consensus, all nodes on a BC must agree on the state of the network, following the predefined rules or protocols. Many consensus protocols have been introduced in BC technology, such as *Proof of Work*, *Proof of Stake*, *Proof of Time*, *Proof of Authority*, etc. However, we will focus on the two most widely used protocols in this paper, i.e., *Proof of Work*, and *Proof of Stake*.

Proof of Work (PoW): First consensus protocol in BlockChain (BC) was proposed by [14] to help participants to agree on a Bitcoin consensus. The consensus protocol relies on a time-consuming calculation involving Hashing (SHA-256), P2P networking, and Merkle Tree algorithm for generating, broadcasting, and checking blocks in the network [14]. PoW introduces the concept of mining which involves validation of a set of transactions (block) in the network PoW incorporates the idea of mining, which entails presenting cryptographic validation of a sequence of transactions (block) in the network through proving the capability of implementing the computational proof of the giving tasks . Once a transaction started, all of the network's miners compete to be the first to solve a cryptographic puzzle to initiate and build the block. The miner who solves the puzzle successfully will broadcast his or her solution (in the form of a block) to other peers over the network. The new block will be formed and approved on the chain after the solution has been checked. Proof of work is a protocol with a primary goal of preventing cyber threats such as DDoS or distributed denial-of-service attacks that aim to drain a computer system's resources by submitting several false requests because each request has a cost i.e., via mining.

Proof of Stake (PoS): Another consensus algorithm is [14], which selects the validator to mine the next block based on its number of coins he/she owns or its stake in the network and the stake age. PoS is implemented with a variety of flavors, ranging from minor to significant modifications to the basic protocol. The most significant distinction between each version of PoS is the strategy each employs to address the protocol's double-spending and centralization issues. To carry out a 51% attack under PoS, the attacker must gain 51% of the participating nodes. In contrast to PoW, an intruder in a PoS scheme is strongly prohibited from initiating a 51% attack because the malicious node fears losing its entire stake if a malicious behavior was repoerted for this node. Both PoS validators (equivalent to miners in PoW) and their stake (cryptocurrency) in the network are tracked by the PoS-based ledger. In a Proof-of-Stake (PoS) scheme, all validators invest stake in the system in order to win the right to mine the next block. If the stake is higher than most of the other stakes, then, the opertunity to be a validator is higher but not guaranteed. The validator for block formation is chosen stochastically by a lottery algorithm that considers validators with higher stakes to have a high probability of winning. If a

node participant wants to cheat the scheme, they will forfeit their share in it. In our work, we use PoS because it does not require any significant computational power and provides a safer network due to the overwhelming attack costs (since the attacker has to acquire 51% of a network's stake tokens).

B. Blockchain Application for Connected Vehicles

Blockchain (BC) is an exciting and versatile technology that has been studied in different application domains. However, few studies have explored using BC in a Connected Vehicle (CV) environment. In this section, we highlight some studies on the deployment of efficient incentive mechanisms and privacy-preservation based on BC technology for CVs. Li et al. [15] introduced CreditCoin, which is an incentive strategy aimed to enhance crowd-sourcing robustness while protecting the privacy of users. When a vehicle notices an unexpected or malicious event, it asks nearby vehicles to check and send back information regarding this irregular event. Once validation, this information is sent to incoming vehicles to warn them to change or modify their trajectories or driving behaviors depending on the existing road conditions. If a vehicle intended to collect traffic data in a certain region, it will provide an incentive or reward on any data transmitted by vehicles in that area. The announcement policy, reward mechanism, and privacy protection were the main three key components of their proposed scheme. They also considered designing a new BC ledger to deal with information verification if needed. However, the implementation complexity, overheads, and security properties of this system are unclear. Singh and Kim [16], [17] use BC to investigate creating of trust and incentive mechanisms for CV. The aim is to provide a framework that allows for safe communication in the CV context without the need for a central authority. They provide a scheme that is based on a BC ledger that allows each vehicle to create a *Bit Trust* which is considered as a unique identifier for each vehicle. Furthermore, BC was utilized to store each vehicle's communication history. A vehicle should contribute to the network's proper functioning in order to receive an incentive and increase its Bit Trust. For example, if a vehicle is engaged with an intersection manager, it will be rewarded and its Trust Bit will be increased by calculating the intersection's crossing order. However, it was not thoroughly investigated if their framework could guarantee the security of knowledge exchanges using BC technologies.

Our work addresses challenges in applying BC consensus mechanisms, such as PoS protocol, to sustain and securely distribute trustworthy scores of CVs. We use a novel design that utilizes a data-driven methodology to detect malicious behavior with decentralized secure infrastructure to track it.

III. Threat Model

The threat model describes our assumptions on the attacker and their capabilities. We consider malicious vehicles that can generate falsified messages and broadcast them to other vehicles. These vehicles are *insider attackers* with previously obtained valid authentication from the *Security Credential Management System* (SCMS)[10]. SCMS is a vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) messaging security solution. To promote trustworthy communication, it implements a Public Key Infrastructure (PKI)-based scheme that uses highly advanced methods of encryption and certificate management. To minimize trackability, SCMS uses *pseudonyms*, which are short-term security credentials created and modified by each vehicle so that it is difficult for eavesdroppers to tell whether BSMs transmitted at the two distinct locations are originated from the same vehicle). According to our attack model, an insider can have access to the on-board unit (OBU) of a vehicle. This malicious vehicle is thought to have the necessary credentials to behave as a legal node, regularly engaging and transmitting false data [18]. Furthermore, the intruder has the capability of changing any field in the BSM components but not to spoof the identities in the messages since this is prevented by the SCMS certificates. In a message spoofing attack, the attacker can send out falsified position and velocity data of itself, which may induce the victim vehicles to accelerate or decelerate. This may degrade the traffic efficiency and even put vehicles near the on-ramp at risk of collisions [18]. The intruder can use a malicious hardware unit or a piece of software to manipulate the transmission rate of its OBU. Furthermore, since it can handle its own OBU, the attacker should be able to use related pseudonyms certificates. We consider only these network based attacks; we do not consider other attacks such as sensor manipulation attacks or physical attacks.

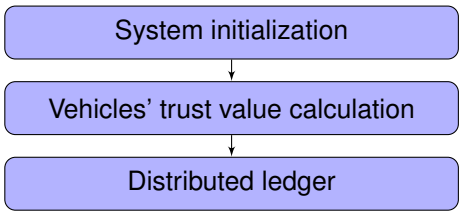

Fig. 1: Flowchart of the proposed system.

IV. Proposed System Architecture

Our proposed scheme includes three phases: *system initialization*, *trust value calculation*, and *distributed ledger construction and maintenance*, as shown in Figure 1. The first phase represents the stage that the connected vehicles gets enrolled and obtains certificates from the Security Credential Management System (SCMS). The second phase is the trust value calculation for each vehicle to measure its reliability. And the last phase refers to the distributed ledger which is shared and consistent via consensus and synchronized to show the recorded vehicular transaction data. The details of all phases are presented in the remainder of this section.

A. System Initialization

A connected vehicle has to obtain a valid certificate before participating in the system. The certificate binds the owner's identity to a pair of encryption keys (i.e., public

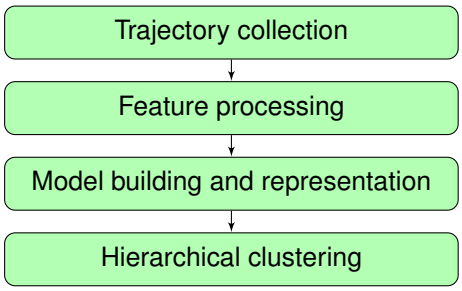

Fig. 2: Flowchart of vehicle trust value calculation.

and private) which are used to encrypt and sign information. Only nodes with valid security certificates and credentials are able to send authenticated messages that will be trusted by the receiving nodes, and participate and contribute to any platform used by the CV system. To obtain certificates, a vehicle has to get enrolled into SCMS by submitting an enrollment request to U.S. Department of Transportation. Once a vehicle is authenticated, it can obtain information such as *Vehicle Trust Values* about other vehicles in the same region through the distributed ledger. Vehicle trust value of the requesting vehicle is updated continuously and can be shared among authenticated entities within the system.

B. Vehicle Trust Value Calculation

This phase is needed to create the vehicle trust estimates based on a data-driven approach to identifying falsified vehicular data. The overview of this phase is shown in Fig.2. This method includes *trajectory acquisition*, *feature extraction*, and *abnormal behavior determination* which is based on an artificial neural network (ANN) model and hierarchical clustering. The details of each step are presented next.

1) Trajectory collection: CV applications mainly rely on basic safety messages (BSMs) which contain dynamic information such as vehicle position, speed, time stamp, acceleration, and other state variables. A trajectory is composed of multiple data instant that reveal information about path behavior over time. Different trajectories are reported by CVs through vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communications, to provide richer spatial and temporal information for better traffic management assessment. Each trajectory data contains BSM data such as location and speed of the vehicle. The selected validator is responsible for using these trajectories to process the Vehicle Trust Value calculation for each vehicle later on. If a vehicle manipulates its information, the Vehicle Trust Value Calculation algorithm will label it with a low trust score which leads to disallowing it from participating in maneuvers.

2) Feature Extraction: Feature extraction is used here to obtain key information from the collected trajectory data for identifying certain patterns that indicate abnormality. In this study, we use three parameters or features that can differentiate various trajectories and help putting them into distinct clusters. These parameters are: a) acceleration rate, b) location index, and c) deferential range. The acceleration rate

Algorithm 1 Creating Features Extraction Table for ANN Model

Input:

▷ We digitize the following parameters for each trajectory into discrete values:

$acceleration\_rate \ \ or\,(A) = \{0,5,10,\dots\}$

$location\_value \ \ or\,(L) = \{1,2,3,\dots\}$

$range\_difference \ \ or\,(R) = \{0,1,2,\dots\}$

$Anomaly\_value\,or \ \ (AV) = \{0,1,2,\dots,10\}$

Output:

$training\_set$

```
Create_Training_Table Function(S, L, R)
 Features ← []
 Index ← 0
 N1 ← length(S)
 N2 ← length(L)
 N3 ← length(R)
 for s_i ← 0 to N1 do
    for l_i ← 1 to N2 do
       for r_i ← 0 to N3 do
          Features[Index] ← s_i + l_i + r_i + AV[Index]
           Index ← Index + 1
return training_set ← Features
```

is a change in velocity for a vehicle through two consecutive time instances and is calculated by dividing vehicle locations by a time difference of two time instances. The location index includes any static position information such as road number, lane number, etc. Finally, deferential range is the space gap between a vehicle and its front vehicle minus the distance measured by the radar sensor. These parameters are inferred using BSMs data to represent each trajectory to be used later on.

In our scheme, the three features are then mapped into discrete ranges, such that defining these trajectories is less computationally demanding and easier to categorize, as shown in algorithm. 1. The trajectory features are fed into the neural network model. The output of the neural network model is a value that will be used in the clustering algorithm to represent a cluster later on.

3) Artificial Neural Network Model: Artificial neural network (ANN) works by having the information from the input neurons multiplied by each weight and then fed into the body layers of the artificial neurons, where these weighted inputs are summed with biases, and transferred through the transfer function to output a final decision. Each artificial neuron can be mathematically modeled as follows:

$$y(k) = f(\sum_{n=0}^{m} w_i(k) * x_i(k) + b_i)$$

where $x_i(k)$ is the input value in the discrete time k and i ranging from 0 to m; $w_i(k)$ is the weight value in the discrete time k; b_i is the bias; f is the transfer function; and $y(k)$ is

the output value in the discrete time k. Our goal is to train the ANN model by having the suitable weights for the hidden layers, so that it can generate the right output based on the right trajectory features.

We use an ANN model with one input layer (three neurons one for each trajectory feature), two hidden layers (64 neurons each) and an output layer that includes 10 neurons for probability distribution of 10 trust credit score values. In our evaluation, we use 80% of the data for training, 10% for testing, and the remaining 10% for cross validation.

4) Trajectory Clustering: In this phase, we apply hierarchical clustering that is an unsupervised learning algorithm that groups similar objects into clusters with similar objects. Each trajectory is treated as an element that is defined by a group of features. These features are used to compute a distance metric to identify the closest cluster to a given element. The falsified trajectory identification can be recognized through having a larger distance from existing clusters of normal trajectories. We use K-means clustering because it is a popular clustering algorithms [19]. K-means works by dividing the data collection into K distinct non-overlapping clusters or subgroups clusters, each of which contains only one data point. The sum of the squared distance between the data points and the cluster's centroid i.e., arithmetic mean of all the data points that belong to that cluster, should be minimal when using K-means to add data points to a cluster.

Within clusters, the less difference there is, the more homogeneous or similar the data points are. The steps to apply the k-means algorithm are as follows (as shown in Fig. 3): (1) The number of clusters, K, is defined. (2) We initialize the centroids by shuffling the dataset first and then choosing K data points at random for the centroids without replacing them; (3) We continue iterating until the centroids do not change; (4) We sum the squared distances between data points and all centroids; (5) we allocate each data point to the nearest cluster (centroid); and (6) We calculate the centroids for the clusters by averaging all the data points that belong to each cluster.

Algorithm 2 Proof of History Algorithm

Input: senderID, regionID, position
Output: Distributed block to next regions
while *A CV in a regoin* **do**
  PoH(*senderID*, *regionID*, *position*)
  if *position* = $region\_boarder_n$ **then**
    A change of region event is created including the updated destination trajectories for the CVs.
    RSU creates and transfers PoH block to the next active regions.
return

C. Distributed Ledger

To increase the trustworthy of the vehicles, we propose adding long term credibility metric for the connected vehicle over time. Thus, we rely on a Proof-of-history (PoH) for verifying vehicles reliability of time between regions. The Proof-of-history credit is mainly responsible of recording vehicles' accumulated spatial and temporal contributions into a ledger. When a vehicle moves across different regions, it is required to update its current active region. This way, the vehicles within the same active region can communicate efficiently. The proof-of-history credit for vehicle i is computed by:

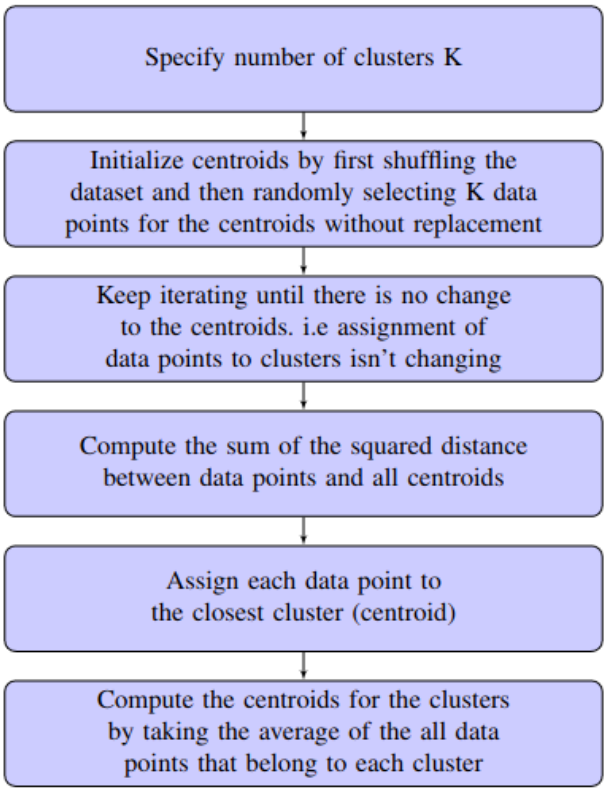

Fig. 3: K-means clustering algorithm.

$$HS_t^i = \sum_{t=t1}^{t2} \alpha * HS_{(t2-t1)}^i \quad (1)$$

Where $HS_{(t2-t1)}^i$ is the proof-of-history credit during the $(t2-t1)$ period; α is a discount factor; and HS_t^i is the accumulated proof-of-history credit during the $t2$ period. This process is shown in Algorithm 2. Note that the credit point of this vehicle in the original region should be set to zero. Once vehicles' trust values are calculated, a distributed ledger can be utilized to identify and expose any abnormal behavior. This distributed ledger provides vehicle trust awareness to other surrounding vehicles, so that the vehicle can use this information before deciding to get enrolled in a certain maneuver or application. The distributed ledger is generated by selected validators that produce BC blocks. Each distributed ledger records vehicles' transaction information such as transaction ID(TID), transaction type(TT), sender ID, credit range, and region ID. In addition, timestamp will be added automatically for each record in the header, which makes it traceable. Then, the distributed ledger validation sender encrypts it with its private key and broadcasts it. To

distribute ledger to other vehicular nodes, validator's election for block generation has to be performed, whose process is shown in Fig.4. Firstly, a vehicle credit is calculated through the equation:

$$vehicle\_credit = HS^i + TS^i + VS^i \tag{2}$$

where HS^i or PoH credit is calculated based on previous credits for the vehicle and can be shared by all the RSU nodes; TS^i or trust score is calculated by the *Vehicle Trust Value Calculation* algorithm; and VS^i or validation score is estimated by measuring the vehicle through different sensors. Then, each vehicle gets a credit range based on its credit value. For example, if the vehicle has a high credit, it gets a range value of 10. If the vehicle has a low credit, it gets a range value as 0. Then, the credit range values of some random vehicles will be collected together in a group or pool. However, this pool includes more vehicles with a high credit range value and less number of vehicles with a lower credit range values which is similar to the concept of POS validator election process. Moreover, a pseudo-random election process will be used to select a validator based on a combination of factors such as the staking age, randomization, and the node's credit range value. Next, the process continues updating validator pool and selecting a validator. A validator has to be elected periodically to manage updating the blockchain due to the decentralized structure of BC technology. The election of a validator ensures the update of data in BC in a timely manner. Finally, the selected validator will be responsible for creating the distributed ledger and broadcasting it.

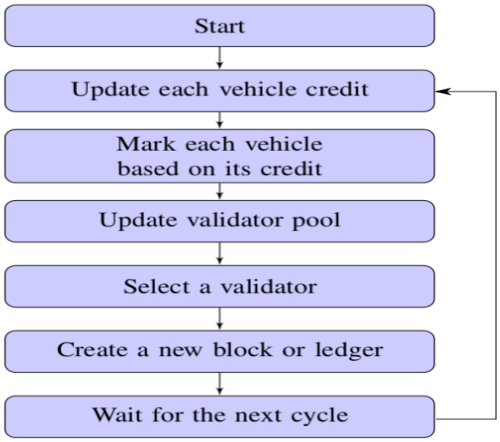

Fig. 4: Validator election process.

V. Cooperative Ramp Merging Algorithm

In this study, we use a cooperative ramp merging application (see Fig. 5) to illustrate the proposed blockchain (BC) technology. In this application, a target connected vehicle (CV) can merge with other CVs from the mainline safely and smoothly through V2X communications. A feedforward/feedback motion control algorithm is developed to obtain the recommended longitudinal acceleration, a_{ref}, [20], which takes into account the target vehicle length l, longitudinal position r, longitudinal speed v, longitudinal acceleration a, and dynamic states from the involved remote vehicles.

$$a_{ref}(t+\delta t) = -\alpha_{ij}k_{ij}\cdot\Big[\Big(r_i(t) - r_j\big(t-\tau_{ij}(t)\big) + l_j + v_i(t) \cdot \big(t^g_{ij}(t) + \tau_{ij}(t)\big)\Big) + \gamma_i \cdot \Big(v_i(t) - v_j\big(t-\tau_{ij}(t)\big)\Big)\Big] \tag{3}$$

where α_{ij} represents the adjacency matrix value; k_{ij} and γ_i are control gains, respectively; The time-varying communication delay between two vehicles is denoted by $\tau_{ij}(t)$; and $t^g_{ij}(t)$ is the time-varying desired time gap between two vehicles. Therefore, the recommended speed can be computed as:

$$v_i(t+\delta t) = v_i(t) + a_{ref}(t+\delta t)\cdot\delta t \tag{4}$$

where $v_i(t+\delta t)$ is the suggested speed; $v_i(t)$ is the current speed of the vehicle; and δt is the length of each time step.

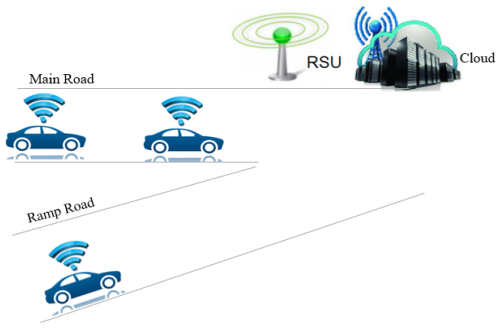

Fig. 5: Cooperative ramp merging scenario

VI. Evaluation

For our experiments, we use VEhicular NeTwork Open Simulator (VENTOS)[21], that is a closed-loop VANET simulator that incorporates communication network and vehicular traffic simulators with a lot of capabilities. It is a free open-source C++ simulator that can create and analyse different traffic flow and intelligent traffic schems, collaborative automated systems, etc. Vehicle-to-everything (V2X) communication can be easly implemented with VENTOS through dedicated short-range communication (DSRC) and other methods. In the simulation, vehicles are generated with Poisson distribution and spawn into a 3-mile network consisting of a 3-lane mainline segment and a single lane on-ramp. We run CVs equipped with DSRC at a maximum speed of 70 mph. The communication range for each vehicle is 300 meters and the roadside units (RSU) is located at the lane merging area. We develop our blockchain (BC) scheme including Transactions to Proof of Stake Consensus in a P2P Network of Nodes in Python as shown in Fig.6.

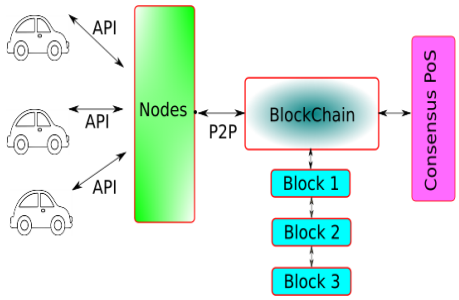

Fig. 6: Proposed blockchain architecture.

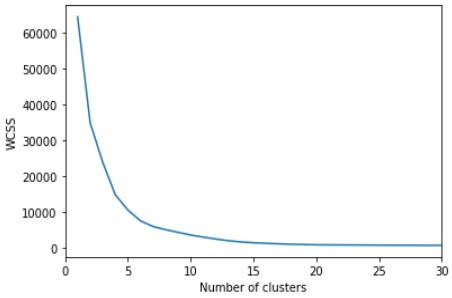

Fig. 9: Elbow method.

In our BC scheme, nodes/vehicles use representational state transfer(REST) API to programmatically query and invoke transactions, and to manage BC network. Our scheme has an Account Balance Model to keep track of the balance of each account as a global state. Fig.7 shows the responding block size based on the number of vehicles in a CV region.

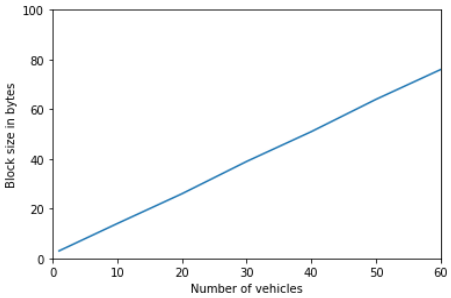

Fig. 7: Block size vs. the number of vehicles.

To show the effects of our attacks, we apply different spoofing attacks to influence the mainline traffic. We measure the total traffic flow for the mainline as shown in Fig.8.

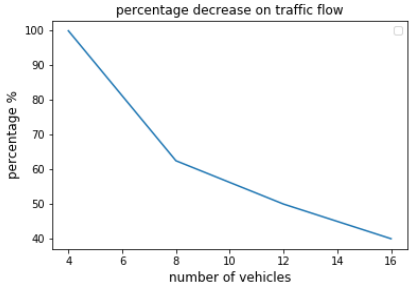

Fig. 8: Attack impact on cooperative ramp merging.

While developing our mitigation scheme, we need to determine the optimal values of system parameters such as cluster number. Thus, we sweep the values of k from 1 to 30. For each k, we compute total within-cluster sum of square (WSS). Then, based on the number of clusters, k, we plot the WSS curve. The position of a bend (knee) in the plot is used to determine the appropriate number of clusters as shown in Fig.9.

Then, we use our simulation to generate normal trajectories based on Newell's car-following model and falsified trajectories to achieve the attacker's goal. Figure 10 shows that the distance between the cluster of falsified trajectory and clusters of other normal trajectories is so significant. This indicates that the proposed clustering method can well identify the falsified trajectory.

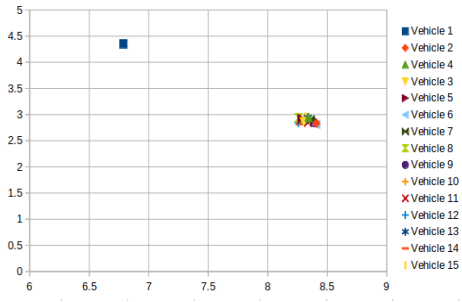

Fig. 10: Clusters' representation for 15 trajectories.

To evaluate efficiency, we measure the execution time for the Vehicle Trust Value Calculation method in our Cooperative Ramp Merging scheme. The results show that this method does not exceed 0.025 seconds as shown in Figure. 11, which indicates the real-time applicability of the proposed method.

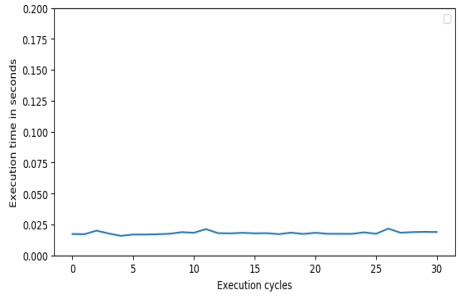

Fig. 11: Time evaluation for vehicle trust evaluation calculation method.

Figure 12 shows the scenario where traffic in the on-ramp margin is under attack. Around the time instant of 30 seconds, the attacker starts its spoofing attacks. If we assume that the forger is selected to start creating the transaction block within the region after one second, then this block will be produced and distributed in less than 2 seconds. To our

best knowledge, this is by far the quickest process compared to other purposed BC technology in intelligent transportation system applications. Therefore, the design of our framework ensures that the attack can be detected immediately and the system can return to the normal condition shortly.

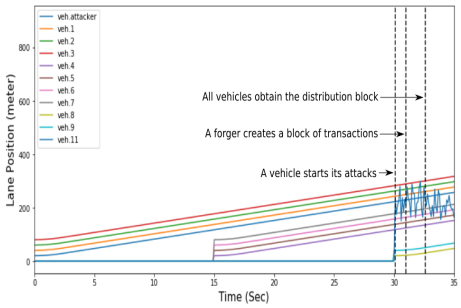

Fig. 12: Effectiveness of our framework against injected attack in the cooperative ramp merging application.

Comparatively, in the scenario of message spoofing attack without the proposed framework, the merging vehicle on ramp will be fooled to speed up so that it creates congestion causing other mainline vehicles to decelerate. This results in degradation of over traffic performance as shown in Table I. We compare both average speed and CO emissions of the merging vehicles under three different cases, i.e., without attack, under attack, and with our framework. The results show that without the protection from our framework, the attack can lead to a 45.3% decrease in average speed and an 21.3% increase in CO emission. Our proposed scheme is able to significantly improve resilience of the system.

Performance	without attacks	with attacks	with Our scheme
average speed (m/s)	6.91	3.78	6.05
CO (mg)	43.66	52.94	43.0

TABLE I: Economic evaluation of our framework against spoofing attacks.

VII. Conclusions and Future Work

In this study, a new architecture for securing connected vehicles is proposed to verify the integrity of other vehicles' messages. Moreover, we propose a solution to addressing the validity of vehicle trajectories using a data-driven approach. Regarding future work, more comprehensive tests (e.g., with hardware-in-the-loop simulation or even in a small scale real world environment) will be performed to verify the proposed blockchain-based security scheme. In addition, we will apply the proposed system to other typical CV applications such as intelligent traffic management and truck platooning, for further evaluation.

Acknowledgments

This work was partially supported by the University of California Office of the President UC Lab Fees grant number LFR-18-548554.

References

[1] J. Siegel, D. Erb, and S. Sarma, "A survey of the connected vehicle landscape–architectures, enabling technologies, applications, and development areas," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–16, 10 2017.

[2] D. Tian, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Performance measurement evaluation framework and co-benefittradeoff analysis for connected and automated vehicles (cav) applications: A survey," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 110–122, 2018.

[3] K. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin, "Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network – performance evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 07 2016.

[4] L. Mendiboure, M. Chalouf, and F. Krief, "Survey on blockchain-based applications in internet of vehicles," *Computers Electrical Engineering*, vol. 84, p. 106646, 06 2020.

[5] ——, "Towards a blockchain-based sd-iov for applications authentication and trust management," 11 2018, pp. 265–277.

[6] C. S. Youngho Park and K.-H. Rhee, "A reliable incentive scheme using bitcoin on cooperative vehicular ad hoc networks," 2017, p. 34–41.

[7] H. security, "Why use a blockchain?" 2020, https://www.hylandcredentials.com/why-use-a-blockchain/.

[8] S. Hu, Q. A. Chen, J. Joung, C. Carlak, Y. Feng, Z. Mao, and H. Liu, "Cvshield: Guarding sensor data in connected vehicle with trusted execution environment," 03 2020, pp. 1–4.

[9] J. Lulka, "Nissan leaf security flaws exposed via hacking," 2016, https://www.digitalengineering247.com/article/nissan-leaf-security-flaws-exposed-via-hacking/.

[10] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications," 2018.

[11] Y. Wang, W. E, W. Tang, D. Tian, G. Lu, and G. Yu, "Automated on-ramp merging control algorithm based on internet-connected vehicles," *IET Intelligent Transport Systems*, vol. 7, no. 4, pp. 371–379, 2013.

[12] J. Peng, H. Huang, and L. Chen, "An adaptive traffic signal control in a connected vehicle environment: A systematic review," *Information (Switzerland)*, vol. 8, 08 2017.

[13] M. Kramer, "What are consensus protocols?" 2019, https://decrypt.co/resources/consensus-protocols-what-are-they-guide-how-to-explainer.

[14] A. Wahab and W. Mehmood, "Survey of consensus protocols," *CoRR*, vol. abs/1810.03357, 2018. [Online]. Available: http://arxiv.org/abs/1810.03357

[15] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[16] M. Singh and S. Kim, "Introduce reward-based intelligent vehicles communication using blockchain," in *2017 International SoC Design Conference (ISOCC)*, 2017, pp. 15–16.

[17] M. Singh and S. Kim, "Intelligent vehicle-trust point: Reward based intelligent vehicle communication using blockchain," 07 2017.

[18] A. Abdo, S. M. B. Malek, Z. Qian, Q. Zhu, M. Barth, and N. Abu-Ghazaleh, "Application level attacks on connected vehicle protocols," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, 2019, pp. 459–471.

[19] P. Patil and A. Karthikeyan, "A survey on k-means clustering for analyzing variation in data," in *Inventive Communication and Computational Technologies*, G. Ranganathan, J. Chen, and Á. Rocha, Eds. Singapore: Springer Singapore, 2020, pp. 317–323.

[20] Z. Wang, K. Han, B. Kim, G. Wu, and M. J. Barth, "Lookup table-based consensus algorithm for real-time longitudinal motion control of connected and automated vehicles," *arXiv:1902.07747v2*, 2019.

[21] M. Amoozadeh, B. Ching, C.-N. Chuah, and D. Ghosal, "Ventos: Vehicular network open simulator with hardware-in-the-loop support," *Procedia Computer Science*, vol. 151, pp. 61–68, 01 2019.