ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838476

A Trust Management Framework for Connected Autonomous Vehicles Using Interaction Provenance

Mohammad Aminul Hoque
Department of Computer Science
University of Alabama at Birmingham
Birmingham, AL 35294-1241
Email: mahoque@uab.edu

Ragib Hasan
Department of Computer Science
University of Alabama at Birmingham
Birmingham, AL 35294-1241
Email: ragib@uab.edu

***Abstract*—Connected autonomous vehicles (CAVs) have fostered the development of intelligent transportation systems that support critical safety information sharing with minimum latency and making driving decisions autonomously. However, the CAV environment is vulnerable to different external and internal attacks. Authorized but malicious entities which provide wrong information impose challenges in preventing internal attacks. An essential requirement for thwarting internal attacks is to identify the trustworthiness of the vehicles. This paper exploits interaction provenance to propose a trust management framework for CAVs that considers both in-vehicle and vehicular network security incidents, supports flexible security policies and ensures privacy. The framework contains an interaction provenance recording and trust management protocol that extracts events from interaction provenance and calculates trustworthiness using fuzzy policies based on the events. Simulation results show that the framework is effective and can be integrated with the CAV stack with minimal computation and communication overhead.**

***Index Terms*—connected vehicles; interaction provenance; trustworthiness; security, misbehaviour detection;**

I. Introduction

Connected autonomous vehicle (CAV) environment supports vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication that enable the exchange of valuable safety and mobility information among the cars and roadside infrastructures. Such communications and autonomous driving are expected to improve traffic conditions significantly [1], [2]. Despite these advantages, the CAV environment imposes significant challenges in secure data transmission among the vehicles to ensure safe and reliable CAV applications [3]. Nearby vehicles may send wrong information intentionally or unintentionally. For example, a vehicle may disseminate a message claiming that a road is clear when there is congestion. Moreover, autonomous vehicles (AVs) may be the victim of a LiDAR spoofing attack that creates a spoofed object, and the vehicle may send messages to other vehicles regarding the obstacle. Hence, evaluating trustworthiness of the vehicles is required because such misbehavior may cause safety consequences.

The IEEE 1609.2 standard [4] defines the formal security requirements in V2V communication that requires all the messages to be digitally signed and verified. Such requirements need a Vehicular Public Key Infrastructure (VPKI). Though such VPKI can efficiently identify outsider attackers, it is challenging to identify inside attackers who possess valid credentials. In this regard, a trust management system maintains the trustworthiness of the participating vehicles considering their past behavior. Trust management can be performed based on a central trusted authority [5] or in a decentralized way by the vehicles and RSUs [6], both having their advantages and disadvantages.

Trust management in the CAV environment imposes several challenges. As the vehicle's activities must be accumulated across multiple pseudonym certificate periods, ensuring the privacy of the cars is essential. Sensors used in AVs are vulnerable to different attack strategies, which also must be considered while calculating trustworthiness. A vehicle revoked due to legitimate misbehaviors can redo the bootstrapping process through Security Credential Management System (SCMS) [1], which is the current standard VPKI. The SCMS does not consider past actions while bootstrapping. Hence, the re-authenticated vehicle can perform similar malicious activities with new certificates. In this circumstance, flexible security policies are required based on the length and quality of the trust management process. Moreover, the mechanism should be able to mitigate collusion and bad-mouthing attacks. Current research works do not support all these requirements together.

In this paper, we propose a trust management framework for the CAV environment using interaction provenance. Interaction provenance is a chronological order of historical interactive events between two or more subjects [7], [8]. Vehicles can rely on the quality of trustworthiness based on the nature and length of previous interactions. In our proposed framework, interaction provenance captures and stores all the in-vehicle and vehicular network events. Such events are used to extract trustworthiness through fuzzy logic engines by creating simplistic policies. Fuzzy ranges and visible contours can significantly benefit the visualization and management of a vehicle's trustworthiness. Other cars, RSUs, or SCMS use the trustworthiness profile in different security aspects. We demonstrate the feasibility of the framework based on an intelligent traffic signal system using VENTOS [9] simulator.

Contribution: The contributions of this paper are as follows:

1) We propose an interaction provenance-based trust management framework for connected autonomous vehicles.
2) We demonstrate the usage of fuzzy control logic with interaction provenance to maintain trustworthiness.
3) We evaluate the framework based on VENTOS simulator to demonstrate the feasibility and effectiveness.

Organization: The rest of the paper is organized as follows: Section II explains the relevant background. Section III contains interaction provenance terminologies. Section IV provides the proposed architecture details. Design discussion in section V is followed by experiment and evaluation in section VI. Section VII presents the related works and we conclude in section VIII.

II. Background

In this section, we provide relevant background regarding Vehicle to Everything (V2X) communication and Security Credential Management System.

A. Vehicle to Everything (V2X) communication

V2V or V2I communication uses Dedicated Short Range Communication (DSRC) or cellular V2X (C-V2X) for exchanging information with low latency and high reliability. DSRC works based on 802.11p physical layer protocol to communicate in the 5.9GHz band over a 10MHz channel. WAVE short message protocol (WSMP) is defined in IEEE 1609.3 as the network and transport layer protocol and IEEE 1609.2 standard defines the security services and protocols for DSRC. Basic Safety Message (BSM) is one of the most important message format defined for V2V communication in the U.S. BSM is exchanged among vehicles typically for 10 times per second.

B. Security Credential Management System

Security Credential Management System (SCMS) is the current standard for VPKI designed by the USDoT. SCMS ensures the vehicle is authenticated and granted enough pseudonym certificates (usually with 5 minutes period) to maintain an anonymous identity. It also removes misbehaving vehicles by revoking the certificates. There are multiple authorities in SCMS for proper registration, certificate distribution, and revocation, which are: enrolment certificate authority (ECA), linkage authority (LA), misbehavior authority (MA), policy generator (PG), pseudonym certificate authority (PCA), device configuration manager (DCM), etc. The MA collects misbehavior reports from multiple vehicles, updates the CRLs, and disseminates the list to revoke the vehicle from the CV environment. The LA provides linkage information to link a vehicle's identity across multiple pseudonym certificates in a privacy-preserving way. LA decides which information can be revealed for linkage purposes.

III. Interaction Provenance for Connected Avutonomous Vehicle environment

In this section, we explain the concept of interaction provenance in the CAV environment. We also demonstrate the applicability of the model for trust management purpose. Figure 1 shows the terminologies and their relations.

A. Interaction provenance terminologies

Actor: In the proposed framework, vehicles, RSUs, and SCMS authorities are considered as the actors who interact with other CAV entities by sending and receiving messages.

Event: An event is a sequence of actions or messages transferred between the actors.

Interaction: An interaction is a message or an action exchanged between two actors during an event.

Interaction provenance: Interaction provenance of an actor is a tamper-proof and chronologically ordered sequence of events.

Interactions are collected based on a particular event or set of events related to vehicular applications. Multiple interactions are comprised in an event. For instance, a roadside equipment collects location and speed information from the nearby vehicles and executes intelligent traffic control algorithm. In this case, the vehicle provides BSM, and retrieves traffic signal information- both of these are events. An event has multiple interactions and its own set of descriptors, which are explained using key-value pairs. A generic structure to represent events and interactions are shown below:

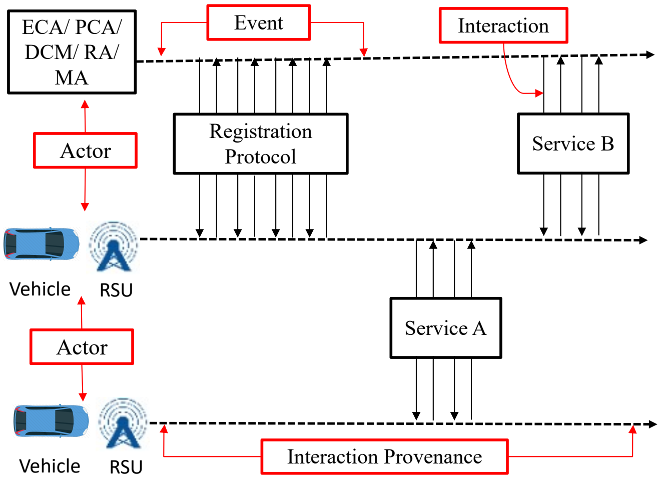

Fig. 1: Interaction provenance terminologies.

EVENT: [Descriptors [key:value], <List: Interactions>]
INTERACTION: [<List: Actor>, Action, EventID, Descriptors[<key:value>]]

B. Trust management using interaction provenance

The framework collects and stores the interactions from the distributed vehicular environment and in-vehicle intrusion detection system (IDS) to calculate trustworthiness. Existing research works [10], [11] can be used as autonomous driving IDS. Interactions are collected in such a way that they contain relevant important information regarding events. Later, the interactions and assigned weights are used to calculate the trustworthiness profile using fuzzy control logic. The trustworthiness profile denotes whether the activities or information provided by the vehicle is historically reliable. Hence, other entities can use trustworthiness information to decide whether to use data from that vehicle in different vehicular applications, calculate the initial reputation score for misbehavior reporting, etc.

IV. System Architecture

In this section, we explain our proposed system architecture. This section contains provenance recording methodology and usage of them for calculating the trustworthiness profile.

A. Interaction provenance frame

An interaction is bound in an interaction provenance frame. The interaction frame consists of two parts: interaction header and payload. The header contains metadata, and the payload consists of the data to be sent or received. Figure 2 shows the structure of an interaction provenance frame. An interaction provenance frame header consists of the following fields:

TS (timestamp), ET (event type), AT (application type), MT (message type), actor1, actor2, actor3, and I-SEQ (interaction sequence). An application may have multiple types of events. For example, the platooning application can have the following events: join, leave, merge, and split. ET can also concatenate multiple event types for an interaction. For example, detected misbehavior or signature verification failure can be concatenated with the actual event type. MT contains the message type used for the interaction. Different message types are defined in the SAE-J2735 standard, such as basic safety message (BSM), Traveler information message (TIM), etc. There can be three actors related to an interaction: actor1 (sender), actor2 (recipient), and actor3. The actor3 field denotes another participant about whom the interaction contains information.

TS	ET	AT	MT	I-SEQ	Actor1	Actor2	Actor3
Interaction Payload							

Fig. 2: Structure of an interaction frame

B. Interaction provenance recording

The interaction provenance recording system consists of Interaction Gateway (IG) service and Trust Management Service (TMS). The IG service collects interactions from the vehicles, RSUs, SCMS entities, and cloud. The TMS receives the interactions from the IG service and maintains the trustworthiness profile for all the CAV entities. The TMS can run as an independent authority or integrated with misbehavior authority (MA) in SCMS. Figure 3 shows the detailed architecture of the proposed framework.

Interaction gateway service: Interaction Gateway (IG) is a distributed service that runs across all the vehicles, RSUs, SCMS entities vehicles and RSUs communicate with, and cloud services related to the CVs. It is responsible for collecting the interactions and reporting to the TMS. The interaction provenance is stored using an XML file. IG collects interactions from in-vehicle intrusion detection system (IDS) and V2X communication. Each event has a descriptor and a list of interactions. The XML file also contains a section for each of the interactions.

The messages are enhanced with interaction headers (IH) in IG to create the interaction frames before sending to the receiver. IG service contains two interaction header handlers (IHH) for processing all the incoming and outgoing messages: incoming IHH and outgoing IHH. The outgoing IHH is responsible for handling all the messages that are going out from the device. Once the message is sent from the vehicle, the interaction parser (IP) of IG prepares and attaches the IH to the application data (payload). IP uses an application database (ADB) and event database (EDB) to create the IH. ADB and EDB contain the mapping of application type and event type to a set of interactions. An example of EDB entries is as follows:

entry e1 = [EVENT_BSM, [POST, /vehicle/bsm]]
entry e2 = [EVENT_SPAT, [GET, /rsu/spat]]

Later, the IP forwards the interaction frame to the outgoing IHH. The outgoing IHH passes the IH through the interaction logger (IL) and sends the message to the vehicular network. Similarly, incoming IHH collects all the incoming messages and checks for the IH. If the receiver field is empty (i.e., the message is broadcasted from the sender), the incoming IHH updates the actor2 field of the header and passes the IH to IL before sending the message to the device. The IL intercepts all the incoming and outgoing interactions and stores them in an interaction database (IDB). Later, the interaction reporter (IR) module periodically collects the interactions from IDB (usually when a pseudonym certificate is about to expire) and signs all of them individually. The signed interactions are encrypted using the public key of trust management authority before sending them to the interaction collector module of TMS.

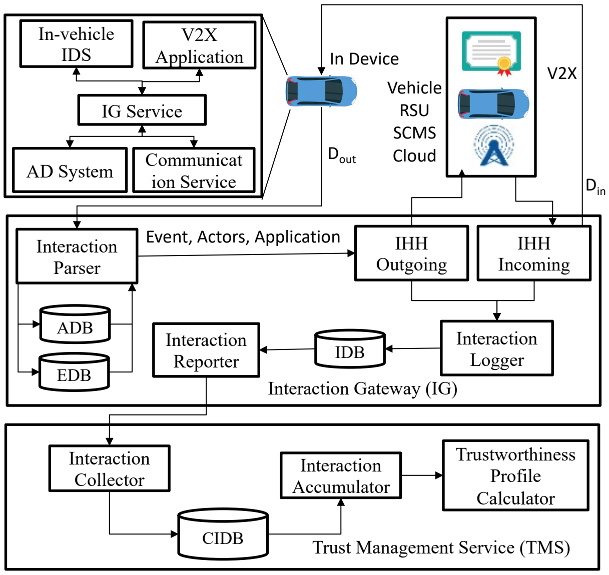

Fig. 3: Architecture of interaction provenance recording framework.

Trust management service (TMS): The TMS is responsible for creating and maintaining the trustworthiness profile of a vehicle based on interaction provenance. The interaction collector (IC) module collects all the interactions from the IR module of IG. IC decrypts the encrypted interactions and verifies the signature. All the interactions are stored in a central interaction database (CIDB) after successful signature verification. The interaction accumulator (IA) module collects and specifies the interactions of a vehicle from CIDB. As the vehicles use pseudonym identity, the TMS contacts with linkage authority (LA) to link all the interactions of a vehicle together. Finally, the trustworthiness profile calculator (TPC) module periodically uses the accumulated interactions to update the vehicle's trustworthiness profile.

C. Weighted interaction and vehicle trustworthiness profile

Each interaction is assigned a weight based on different properties. We specify two properties of an interaction and their weight assignment considerations to determine the aggregated weight of the interaction.

Event (W_1): Weight is specified to the interaction based on the event type for a particular application. For instance, interactions for registration, certificate revocation, or misbehavior report events receive higher weights. On the other hand, the events related to platooning or traffic signal control applications receive smaller weights. Depending on the event and application type, the interaction can have the following weights: very high, high, low, and very low. Examples of such event types include:

- **Very high:** Misbehaviour report, Certificate revocation, Signature verification failure
- **High:** Registration
- **Low:** Forward collision warning
- **Very low:** platooning, intelligent traffic signal control

Time (W_2): The weight of an interaction is specified based on the event time, which reduces with the increasing difference with the current time. Hence, recent events have larger weights, and older events receive smaller weights.

Fuzzy ranges of the weights: We define four weight mappings for each properties of an interaction: very high, high, low, and very low. All the weight assignments are performed in a pre-specified scale. We specify the following fuzzy ranges between [0-1] for the input fuzzifier and output defuzzifier: (a) **very low: [0.0-0.35]**, (b) **low: [0.2-0.6]** , (c) **high: [0.35-0.75]**, and (d) **very high: [0.6-1.0]**.

Aggregated weight of interaction: Each of interaction properties are used with a combination function to calculate the $weight$ of an interaction. The $interaction\ weight$ is calculated as follows:

$$InteractionWeight = \alpha_{E1}W_1 + \alpha_{E2}W_2 \tag{1}$$

Here α_x is a scalar coefficient for the linear combination function and $x\epsilon\{E1, E2\}$. For vector aggregation, polynomial feature can also be used to create grater impact with increasing weight for a specific property. Multiple mathematical model for vector combination exist such as normalized summation, simple algebraic, bounded, Hamacher, etc.

Vehicle trustworthiness profile mapping: A defuzzifier function is designed based on a scale of predefined ranges similar to the fuzzifier function and applied to the aggregated weight. The defuzzifier provides a trustworthiness mapping for a particular vehicle. Common approaches for defuzzification include weighted average, weighted sum, bisector, largest, mean of maximum, and so on. The access granted to a vehicle in the, initial reputation score, or similar decisions can be taken based on the trustworthiness mapping provided by the defuzzifier.

Fuzzy control logic: We use a fuzzy range to classify the input and output ranges, which was mentioned earlier. The fuzzy control logic allows simple and easy to understand linguistic rules to define the trustworthiness of a vehicle. The defuzzified trustworthiness is achieved by the control logic and fuzzy ranges. Later, the trustworthiness is evaluated against threshold mapping. Some examples of fuzzy control logics are:

- If event is **very high** and time is **low**, then trustworthiness is **low**.
- If event is **very high** and time is **very high**, then trustworthiness is **very low**.
- If event is **very low** and time is **high**, then trustworthiness is **very high**.
- If event is **very high** and time is **low**, then trustworthiness is **high**.

D. Dissemination of updated trustworthiness profile

TMS uses the newly arrived interaction provenance data and updates the trustworthiness profile using fuzzy policies. Later, it disseminates the profile to the RSUs. Hence, all the RSUs hold the same trust profile for a vehicle. The vehicles can request for updated trust score from RSUs based on the current pseudonym identity.

V. Design Discussion

In this section, we analyze the design of the proposed framework. Interaction provenance holds the events that a vehicle encounters in a chronological sequence. Moreover, the provenance also captures events from both in-vehicle and vehicular networks. Here we analyze the proposed design:

Autonomous driving security events: Autonomy is going to be an integral part of vehicles in the near future. Hence, managing trust only based on vehicle connectivity does not provide complete trustworthiness. The proposed framework supports consideration of security events of both connectivity and autonomy of the vehicle to calculate trustworthiness, which is not supported by current research works.

Privacy and linkability: According to SCMS, the vehicles change their pseudonym certificates after sometimes (typically 5 minutes). Hence, it would not be possible to accumulate trustworthiness across multiple pseudonym certificate periods in decentralized trust management schemes unless the Linkage Authority (LA) shares linkage information with RSUs or base stations. However, SCMS currently does not support this to ensure privacy. Though several works provide linkability [12], they do not comply with SCMS. The proposed framework directly cooperates with LA and does not reveal private information; providing required privacy and linkability.

Security against bad mouthing: The framework is capable of preventing bad mouthing (wrong reputation dissemination regarding a benign vehicle). For this purpose, the framework can exploit the Bayesian inference technique similar to previous works [6], [13] to identify whether any false reputation message has been disseminated.

Security against collusion attacks: Several vehicles may collude among themselves to manipulate trustworthiness by exchanging benign messages. Interaction provenance is highly helpful to prevent such collusion scenarios. The fuzzy policies can be adjusted to emphasize interactions with multiple different CAV entities to update trustworthiness. Weights of interactions among the same actors can be reduced gradually. Hence, the colluding vehicles will not be able to manipulate the trustworthiness of a vehicle significantly.

Flexible security policies: The trustworthiness profile of the vehicle can be used to design flexible security policies. For example, misbehavior detection techniques initialize the reputation point from a fixed value irrespective of its trustworthiness. Moreover, granted access to a vehicle in the CAV environment also does not depend on trustworthiness. For these purposes, *Adaptive reputation initialization* or *threshold access control* can be achieved based on the length and quality of the events the vehicle encountered previously.

VI. Experiment and Evaluation

In this section, we explain the experiment and evaluation of the proposed framework. We implement a prototype and analyze the performance of our proposed framework based on intelligent traffic control system.

A. *Intelligent traffic signal control system*

The intelligent traffic signal system is operated on an RSU located in an intersection. The vehicles within the DSRC communication range send location and velocity information encoded in BSM to the RSU. The RSU executes the traffic signal control algorithm using different heuristics to calculate the estimated arrival time of the vehicles. The traffic signal controller optimizes the waiting time and updates traffic lights accordingly based on the estimation. There can be a total of eight traffic signal controls in the intersection shown in figure 4, which are called phases. Each traffic light state can be green, red, or yellow. The green state varies based on the output of the algorithm that optimizes the wait time.

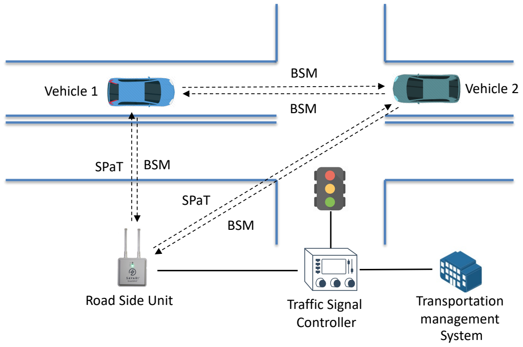

Fig. 4: Workflow of the intelligent traffic control system

B. *Experiment setup*

For experimental evaluation of the proposed framework, we used VENTOS [9] - an open-source vehicular networking simulator developed on top of SUMO [14] and Omnet++. We created a four-way intersection where the vehicles could move forward, turn left, and turn right. We placed an RSU in the middle of the intersection that could receive BSMs and respond with SPaT regarding signal states. For the cryptographic operations, we used SHA-256 hashing and the RSA-2048 signature algorithms.

We implemented a prototype of the proposed framework that created and stored interaction provenance for all the intelligent traffic control system entities. We assumed all the vehicles and RSUs were bootstrapped and authenticated by different CAs of SCMS. In the simulation, all vehicles within the communication range exchanged messages with RSUs. One of the vehicles acted maliciously and intentionally sent forged BSMs for manipulating the signal control algorithm. The RSU properly executed the misbehavior detection algorithm to figure out the vehicle and reported it to the misbehavior authority. Hence, the following events could occur in our implemented prototype: registration, sending BSMs, receiving SPaT, misbehavior report, and certificate revocation. All these interactions were collected and stored in a Sqlite database.

C. *Trust management based on proposed framework*

We used *fuzzylite* [15] to create fuzzy logic for vehicle trustworthiness profile, which is an open-source library for fuzzy control logic. For vector aggregation, we considered the event type as the most crucial property of the interaction, followed by the time property. We created 60 fuzzy rules and generated a contour map. For this purpose, we used four fuzzy range values for input/output (very high, high, low, and very low) and a Gaussian model of variance 0.05 for each range. In our defined weight mapping, higher event weight corresponded to certificate revocation, misbehavior reported, etc. Such mapping policy leads to lower trustworthiness of the device involved with such events. Hence, the vehicle with higher event weight received lower weight mapping by the output defuzzifier and was defined as less trustworthy. On the other hand, message exchange events for regular applications were assigned lower weights. We calculated a fuzzy contour map that provides a 2-dimensional visual representation of vehicle trustworthiness profile. Such visualization of contour maps allows the evaluation of the fuzzy policies easily. Figure 5 shows the contour map for vehicle trustworthiness profile. The darker green color represented the vehicle as more trustworthy, which reduced with the movement of contour towards lighter colors. We defined our rules so that the vehicle containing malicious or negative event history in interaction provenance would never receive higher trustworthiness unless they became too old to be considered.

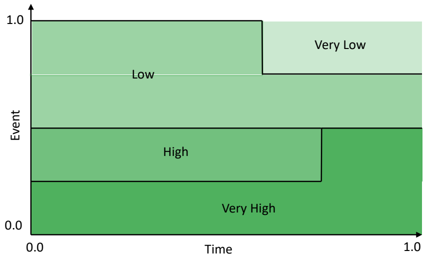

Fig. 5: Fuzzy contour map for vehicle trustworthiness profile

D. *Communication and computation overhead*

Here, we present the experimental results in terms of communication and computation overhead.

Message overhead: In the proposed mechanism, the message overhead was the added interaction header with the application data. Here, the size of the interaction header was fixed, which was calculated as follows: TS ← 4 bytes, ET ← 1 byte, AT ← 4 byte, MT ← 4 bytes, actor1 ← 16 bytes, actor2 ← 16 bytes, and actor3 ← 16 bytes, I-SEQ ← 4 bytes: a total of 65 bytes. We assigned 16 bytes for actor identification which was equal to IPv6 address. Figure 6 shows the percentage of message overhead. Due to the constant size of the header, the overhead was higher for smaller messages which decreased with increasing message size. Hence, the message overhead was minimal for larger messages.

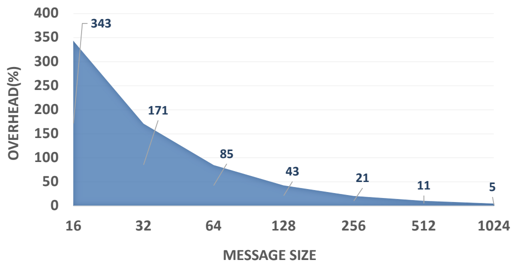

Fig. 6: Percentage of overhead with varying message size

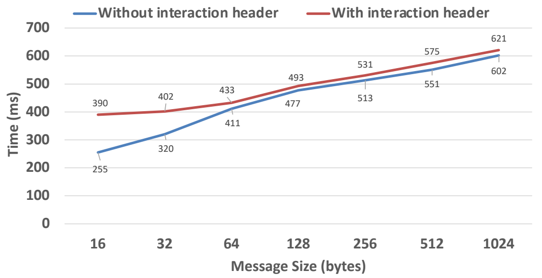

Fig. 7: Computation overhead with varying message size

Computation overhead: We calculated the time to send the message with and without the interaction frame to quantify the computation overhead. For the baseline, we calculated the message sending time without the overhead created by the framework. Later, we measured the required times for database access, interaction frame creation, and sending the message along with the interaction header. Figure 7 shows the overhead in milliseconds for sending the message. We observe that overhead got reduced with increasing message size.

VII. Related works

Various trust management mechanisms for the CAV environment have been proposed in the literature. Blockchain has become a popular technique in this regard [6], [13], [16]. However, these techniques require to reveal identification information, such as VIN [6] or vehicle id [13], [16] which lead to privacy issues. Centralized trust management [5], [17] is another approach where a central trusted authority manages trusts for all the CAV entities. Here, none of the research works considered both connectivity and autonomy of the vehicle together for trust management. Reputation and misbehavior detection-based trust management has also gained significant research attention. Different such techniques include machine-learning-based detection [18], [19], context-aware trust management [20], Kalman filter-based approach [21], etc. Trust management in vehicular social network has also been proposed based on reputation score [22]. However, these schemes consider activities for one pseudonym certificate cycle and do not consider long-term activities.

Interaction provenance has been used for different purposes in the literature. Khan *et al.* [7] proposed an interaction provenance-based authentication and access control for service-oriented computing. Hossain *et al.* [23] proposed a forensic investigation framework for the internet of vehicles. Usage of fuzzy logic also exists in the literature for connected vehicle environment [24]. Though interaction provenance and fuzzy logic technique has been used in different paradigms, to the best of our knowledge, ours is the first attempt to utilize them for trust management for CAVs.

VIII. Conclusion

The CAV environment is vulnerable to different internal attacks where falsified data can affect the applications. A trust management process maintains the vehicles' trustworthiness and decides whether to use the data shared from the vehicle. In this paper, we exploit the causality of social interactions with other subjects for trust management. The proposed framework uses interaction provenance that represents the vehicle's activities. We also show how fuzzy logic can be used to create the trustworthiness profile of a vehicle. We have implemented a prototype and demonstrated the feasibility of the framework based on an intelligent traffic control application.

Acknowledgement

This research was supported by the National Science Foundation through awards ACI-1642078 and CNS-1351038.

References

[1] B. Brecht and T. Hehn, "A security credential management system for v2x communications," in *Connected Vehicles*. Springer, 2019, pp. 83–115.

[2] USDoT, "Connected vehicle pilot deployment program," 2019. [Online]. Available: https://tinyurl.com/e94jnw5n

[3] M. A. Hoque and R. Hasan, "Towards an analysis of the architecture, security, and privacy issues in vehicular fog computing," in *2019 SoutheastCon*. IEEE, 2019, pp. 1–8.

[4] "Wireless access in vehicular environments–security services for applications and management messages," *IEEE Standard 1609.2-2016*, 2016.

[5] C. Lai and X. Shen, "Sirc: A secure incentive scheme for reliable cooperative downloading in highway vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1559–1574, 2016.

[6] Z. Yang, "Blockchain-based decentralized trust management in vehicular networks," *IEEE IoT Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[7] R. Khan and R. Hasan, "Fuzzy authentication using interaction provenance in service oriented computing," in *2015 IEEE International Conference on Services Computing*. IEEE, 2015, pp. 170–177.

[8] M. A. Hoque and R. Hasan, "An interaction provenance-based trust management scheme for connected vehicles," in *IEEE CCNC*. IEEE, 2022, pp. 731–732.

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

[10] Y. Wang and D. Wang, "Detection and isolation of sensor attacks for autonomous vehicles: Framework, algorithms, and validation," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] M. A. Hoque, M. Hossain, and R. Hasan, "Benchav: A security benchmarking framework for autonomous driving," in *IEEE CCNC*. IEEE, 2022, pp. 729–730.

[12] T. Pham and C. Yeo, "Adaptive trust and privacy management framework for vehicular networks," *Vehicular Comm.*, vol. 13, pp. 1–12, 2018.

[13] X. Chen, J. Ding, and Z. Lu, "A decentralized trust management system for intelligent transportation environments," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[14] SUMO, "Simulation of urban mobility," 2021. [Online]. Available: https://www.eclipse.org/sumo/

[15] "The fuzzylite libraries for fuzzy logic control," 2021. [Online]. Available: https://www.fuzzylite.com/

[16] F. Kandah, B. Huber, A. Skjellum, and A. Altarawneh, "A blockchain-based trust management approach for connected autonomous vehicles in smart cities," in *CCWC*. IEEE, 2019, pp. 0544–0549.

[17] M. E. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, 2011.

[18] P. Sharma and H. Liu, "A machine-learning-based data-centric misbehavior detection model for internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, 2020.

[19] S. Gyawali and R. Q. Hu, "Machine learning and reputation based misbehavior detection in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, pp. 8871–8885, 2020.

[20] J. Guo, X. Li, Z. Liu, J. Ma, C. Yang, J. Zhang, and D. Wu, "Trove: A context-awareness trust model for vanets using reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6647–6662, 2020.

[21] M. Sun, "A data trust framework for vanets enabling false data detection and secure vehicle tracking," in *IEEE CNS*. IEEE, 2017, pp. 1–9.

[22] N. Ullah, X. Kong, Z. Ning, A. Tolba, M. Alrashoud, and F. Xia, "Emergency warning messages dissemination in vehicular social networks: A trust based scheme," *Vehicular Comm.*, vol. 22, p. 100199, 2020.

[23] M. Hossain and R. Hasan, "Trust-iov: A trustworthy forensic investigation framework for the internet of vehicles," in *ICIOT*, 2017, pp. 25–32.

[24] H. Xia, S.-s. Zhang, B.-x. Li, L. Li, and X.-g. Cheng, "Towards a novel trust-based multicast routing for vanets," *Security and Communication Networks*, vol. 2018, 2018.