2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00117

NWADE: A Neighborhood Watch Mechanism for Attack Detection and Evacuation in Autonomous Intersection Management

Jian Kang
University of Missouri
jkang@mail.missouri.edu

Alian Yu
University of Missouri
ay9mm@missouri.edu

Wei Jiang
University of Missouri
wjiang@missouri.edu

Dan Lin
University of Missouri
lindan@missouri.edu

***Abstract*—With the advances in autonomous vehicles and intelligent intersection management systems, traffic lights may be replaced by optimal travel plans calculated for each passing vehicle in the future. While these technological advancements are envisioned to greatly improve travel efficiency, they are still facing various challenging security hurdles since even a single deviation of a vehicle from its assigned travel plan could cause a serious accident if the surrounding vehicles do not take necessary actions in a timely manner. In this paper, we propose a novel security mechanism namely NWADE which can be integrated into existing autonomous intersection management systems to help detect malicious vehicle behavior and generate evacuation plans. In the NWADE mechanism, we introduce the neighborhood watch concept whereby each vehicle around the intersection will serve as a watcher to report or verify the abnormal behavior of any nearby vehicle and the intersection manager. We propose a blockchain-based verification framework to guarantee the integrity and trustworthiness of the individual travel plans optimized for the entire intersection. We have conducted extensive experimental studies on various traffic scenarios, and the experimental results demonstrate the practicality, effectiveness, and efficiency of our mechanism.**

***Index Terms*—Autonomous vehicles, intersection management, attack detection and evacuation, blockchain.**

I. Introduction

You may be surprised that the amount of time a person spent at intersections is astoundingly 58.6 hours on average each year [8]. Such a huge waste of time and energy may be eliminated in the near future with the fast growth in autonomous vehicles. Various intelligent traffic management systems have been proposed that can assign travel plans to individual vehicles to help them cross intersections without much stop [7], [10], [12], [16], [25], [27], [28], [30], [35], [38]–[40].

Although the above envision is very attractive, security and safety present the greatest challenge for such systems to be successfully deployed in the real world [17]. Imagine that a compromised or malfunctioned vehicle left its designated lane or made an unexpected turn, it could easily cause a series of collisions and casualties. Even worse, if the central unit (i.e., the intersection manager) of an intelligent traffic management system is hacked, the travel plans generated by the intersection manager could directly lead to chaos and disasters at the intersection if no security measure is in place [6], [21].

As initial efforts towards the security and safety protection for the intelligent traffic management system, there have been protocols for authenticating vehicles [14], [24], [41], authenticating messages sent by vehicles [1], [3], [26], and determining message receivers [19], [20], [22], [29]. However, these methods are far from sufficient to defend against zero-day attacks. For example, the attacker may still be able to take control of a vehicle and cause an accident after the vehicle has successfully authenticated. In other words, authentication and access control mechanisms may be useful to identify the problematic vehicle after the accident, but cannot prevent the compromised vehicle from conducting malicious behavior on the scene. Also, some misbehavior may be due to unexpected mechanical or technical problems of the vehicle, and such misbehavior cannot be prevented by existing security protocols either. Moreover, only detecting the nearby vehicles' abnormal behaviors is not sufficient to prevent the accident from happening. A well-designed mechanism is necessary to evaluate the trustworthiness of the detection report sent by the witness vehicles.

In order to address the aforementioned challenges, we propose a novel mechanism called Neighborhood Watch for Attack Detection and Evacuation (NWADE) to provide security guarantees for intelligent intersection traffic management systems in both big cities with high vehicle densities and small towns with low vehicle densities. In the NWADE system, an intersection manager generates travel plans to schedule incoming vehicles to cross the intersection. The interesting idea underlying the NWADE mechanism resembles the neighborhood watch practice in our daily life. With our NWADE mechanism in the system, each vehicle will keep an eye on surrounding vehicles by utilizing its equipped sensors and onboard computing units. The biggest challenge of the design lies in combating a series of complicated threat scenarios imposed by the following questions:

- Can the incoming vehicles trust the travel plans generated by the intersection manager? How can they know the plan would not cause collisions?
- Can the intersection manager trust the vehicle which reports spotting a malicious vehicle nearby?
- Will a majority vote of vehicles surrounding a suspicious

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00117

vehicle be sufficient to confirm or clear the alarm? What if there are a group of malicious vehicles traveling together to game the majority vote scheme?
- Can the vehicles trust the evacuation plan broadcasted by the intersection manager or other peer vehicles? What if that is just a sham because the intersection manager or the peer vehicle has been compromised?

As any party including the intersection manager and one or more vehicles in the system may be compromised, correctly determining when and what action should be taken by each party is crucial to the safety of the entire system. Our contributions to this work are summarized as follows:

- We propose a sophisticated attack detection mechanism that can not only identify but also validate a vehicle's misbehavior in real-time. Our detection mechanism is built upon innovative collaboration protocols among peer vehicles.
- We leverage the blockchain techniques to ensure the integrity and consistency of travel plans generated and disseminated by the intersection manager, which serves as the fundamental building block for the attack detection mechanism.
- We have conducted an in-depth analysis of various scenarios that could render security threats, and our proposed NWADE mechanism demonstrates robustness in all cases. Specifically, false incident reports will be detected while true incident reports will activate evacuation plans.
- Our proposed NWADE mechanism can be integrated into most existing intelligent intersection management systems. We have extensively evaluated the NWADE mechanism under a variety of intersections and traffic flows with different intersection management systems. The experimental results show that our proposed security mechanism is very efficient and can be used in highly dynamic and time-sensitive environments.

The rest of this paper is organized as follows. Section II reviews related works. Section III presents the motivation and threat model. Section IV introduces our proposed NWADE mechanism. The security analysis and experimental results are presented in Section V and VI, respectively. Lastly, Section VII concludes the paper.

II. Related Work

The efforts toward the security and safety protection for the intelligent traffic management system can be classified into three main categories: (i) Authentication; (ii) Authorization; and (iii) Message trustworthiness verification. It is worth noting that none of these existing approaches can prevent all the attacks as discussed in our threat model. They are complementary to our work to help hold the malicious vehicles accountable after the incidents.

The vehicular authentication protocols allow vehicles to check whether a peer vehicle has a legitimate registration, i.e., a verified identity. The identity could be either real or anonymous [14], [15], [18], [33], [42]. Some authentication protocols also employ the blockchain techniques [5]. However, the use of the blockchain is totally different from our work. Authentication provides the first defense against malicious vehicles without valid identities, but authentication protocols alone including blockchain-based authentication are far from sufficient to guarantee road safety. This is because authentication does not prevent an authenticated vehicle from being compromised by an attacker, and performing attacks on other vehicles under the attacker's control.

The authorization mechanisms allow vehicles on the roads to designate a group of vehicles to access their messages [22], [29], [36]. This is more for privacy protection rather than safety protection as discussed in our work.

In terms of message trustworthiness validation, Most of the existing works on this topic are developed based on the idea of reputation systems [9], [11], [26]. Such reputation-based systems will not be able to prevent attackers from exploiting a compromised vehicle that already has a high reputation to send false reports.

There are also some efforts on detecting abnormal behaviors in intelligent traffic management systems. Kremer et al. [23] propose a state estimator to detect malicious activities within a vehicle platoon. The limitation is that it cannot detect whether multiple groups of vehicles from different incoming lanes may collide at the intersection. Heijden et al. in [34] reviewed existing schemes that can detect misbehavior among vehicles and evaluated the correctness of the information. However, these schemes focus on detecting malicious messages rather than malicious driving behavior. They cannot prevent attackers, which do not need to send any malicious messages, from maneuvering compromised vehicles to cause a collision.

Most recently, some blockchain-based message validation approaches have been proposed [3], [4], [13], [32]. For example, Buzachis et al. [4] propose to utilize the blockchain and smart contracts to ensure the integrity of the data. Rathee et al. [32] propose to utilize the blockchain to record every activity of the vehicles and auditing purposes after the accidents occurred. However, it is not sufficient to prevent compromised vehicles from launching attacks. At the first look, these existing blockchain-based message validation approaches may seem similar to our blockchain verification. However, they are indeed very different. Existing works utilize the blockchain to verify the owners of messages. They are not able to validate the content of the messages. For example, a vehicle hacked by the attacker will pass the message validation by the existing approaches even if the compromised vehicles are reporting false incidents. A compromised intersection manager can send conflicting travel plans without being detected by the existing message validation approaches as well. To sum up, none of the existing works can handle all the attack scenarios discussed in our work.

III. Motivation and Threat Model

In this work, we are focusing on protecting the future intelligent intersection management systems which consist of two parties: (i) autonomous vehicles; and (ii) intersection manager

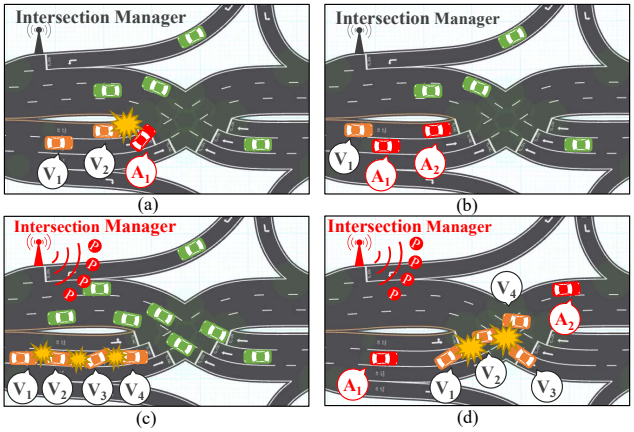

Fig. 1. Security Threats

(or called road-side unit). We assume that the vehicles can communicate with each other using existing VANETs (Vehicular Ad-hoc NETworks) protocols or 5G technologies. There are various existing intelligent intersection management systems that can be used for scheduling autonomous vehicles such as optimal traffic light scheduling [40], platoon-based scheduling [37], [39] and motion-planning [16], [38]. Their common goal is to find the optimal travel plans for incoming vehicles to cross the intersection as fast as possible. Specifically, when a vehicle approaches an intersection, it sends its status such as speed and turning direction to the intersection manager. The intersection manager constantly monitors the overall traffic at the intersection, calculates the optimal scheduling, and sends the travel plans to individual vehicles.

Unlike previous works on the vehicular network security [22], [24], [31], [36], [43] that typically assume the intersection manager to be trustworthy but curious to ease the security protocol design, our work aims to conquer a wider range of security threats that may happen in the real world. We classify the possible threats into four categories in ascending order of the attackers' capabilities:

(i) A single vehicle behaves maliciously. A problematic vehicle may violate the travel plan assigned by the intersection manager. The deviations from the travel plans, such as moving faster or pressing the brake, may cause a series of chain reactions. For example, as illustrated in Fig. 1 (a), the malicious vehicle A_1 made a sudden lane change and collided with the vehicle V_2.

(ii) Multiple vehicles have been compromised. This case is more challenging which assumes the attacker has abilities to hack more than one vehicle. The compromised vehicles may even be close to each other to conduct collaborative attacks. Fig. 1 (b) shows an example where two malicious vehicles A_1 and A_2 intentionally block the benign vehicle V_1's route. Moreover, if the malicious vehicles outnumber the benign vehicles at a road segment, the majority-voting-based message verification may be exploited by malicious vehicles, and true incident reports sent by benign vehicles may be voted as false.

(iii) The intersection manager has been compromised. Although the intersection manager is likely to have stronger security protection, it does not mean it will be free from attacks. The intersection manager schedules all vehicles' travel plans and can wreak more damage if it is taken control by the attacker. For example, as shown in Fig. 1 (c), a malicious intersection manager may send out wrong travel plans (P) to induce pile-up accidents. Thus, it is important to detect such attacks and have a backup plan to guarantee vehicles' safety in case the intersection manager is compromised.

(iv) Intersection manager and multiple vehicles have been compromised This could be the most challenging scenario when the attacker gains control of the intersection manager and several vehicles at the intersection. As shown in Fig. 1 (d), the attacker may plan a larger scale of car accidents by exploiting the intersection manager's scheduling abilities and by using malicious vehicles to disseminate false traffic situations to mislead normal vehicles.

To sum up, the ultimate security goal of our system is to avoid accidents while maximizing travel efficiency. The scope of this work focuses on the data integrity problem as the first step toward the safety of autonomous vehicle scheduling. Regarding privacy concerns, some lightweight privacy protection approaches [14], [18], [31], [42] may be integrated into our system. Moreover, authentication mechanisms for autonomous vehicles are complementary our system as well.

IV. A Neighborhood Watch Mechanism for Attack Detection and Evacuation in Autonomous Intersection Management

The NWADE (Neighborhood Watch for Attack Detection and Evacuation) mechanism aims to help existing intelligent traffic management systems to mitigate the security threats. Specifically, the intersection manager will use a blockchain to store the travel plan of each incoming vehicle to ensure the integrity of the travel plans. The vehicle in an intersection will request from the intersection manager not just its own travel plan but also several blocks of previous vehicles' travel plans for verification purposes. Each vehicle will also serve as a local verifier (or watcher) to monitor surrounding vehicles' movements by using equipped sensors and report any abnormal behavior. If a vehicle detects abnormal behavior of the intersection manager, it will broadcast a global report to warn other vehicles. Based on the information received from peer vehicles and the intersection manager, each vehicle will make informed decisions to protect its own safety.

A. Event-driven Deterministic Finite Automation in NWADE System

The key challenge of the design is to have sophisticated protocols that enable individual vehicles to make correct judgments regarding the information received from different channels including the intersection manager, the local verifiers, the global verifiers, and even the malicious vehicles. In order to model the complicated interactions among vehicles and the intersection manager, we build event-driven deterministic

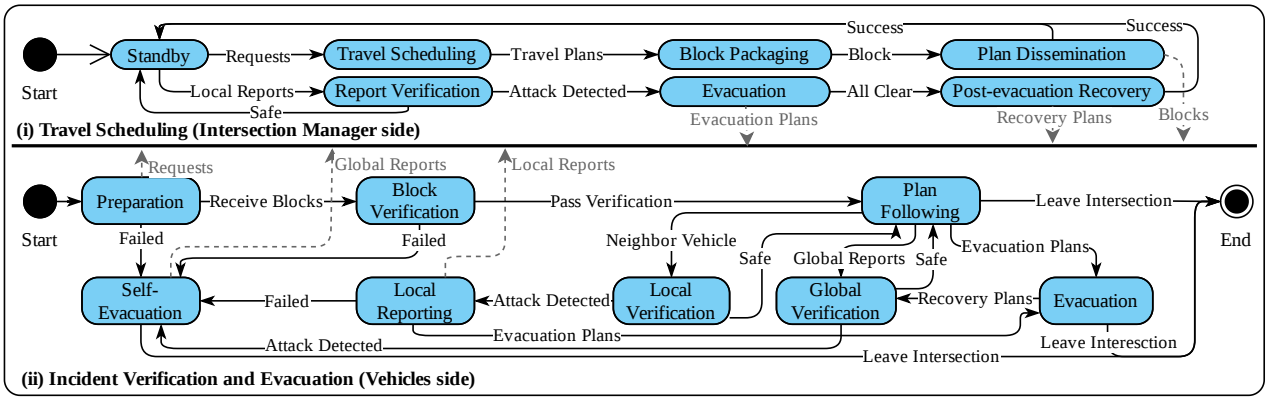

Fig. 2. Event-driven Deterministic Finite Automation in NWADE System

finite automata for vehicles and the intersection manager, respectively.

At a given time point, the intersection manager may be at one of the 7 states as shown in Fig. 2 which model the following two major tasks:

(i) Travel Scheduling (Intersection Manager side): In the beginning, the system is at the standby status. Upon receiving the requests from vehicles, the intersection manager will enter the travel scheduling stage. The travel plans will be calculated depending on the specific algorithms adopted by different intelligent traffic management systems. Once travel plans are generated, the NWADE mechanism will bring the intersection manager into the block packaging stage in which the intersection manager packages the newly generated travel plans using the blockchain technique (as presented in Section IV-B1). Then, the blocks containing the travel plans will be disseminated to the vehicles. After that, the intersection manager will return to the standby status to handle the next requests.

(ii) Incident Verification and Evacuation (Vehicles side): A vehicle may notify the intersection manager when spotting its neighboring vehicle's suspicious behavior. Upon receiving such an incident report, the intersection manager will execute the report verification process (as elaborated in Section IV-B2). If the report is a false alarm, the intersection manager simply goes back to the standby status. If the threat is confirmed, the intersection manager will enter the evacuation phase to generate evacuation plans for vehicles to leave the intersection safely. Once the threat is cleared, it will enter the post-evacuation recovery stage to resume the traffic by recalculating the travel plans based on the vehicles' status at that moment.

As for individual vehicles, they may enter any of the 8 states (Fig. 2) when they are passing the intersection depending on the real-time traffic situation. The 8 states aim to model the following 4 tasks a vehicle may need to perform to ensure their own safety:

- **Normal Traveling**: When a vehicle enters the communication zone of the intersection manager, the preparation phase begins whereby the vehicle will send its dynamic information to the intersection manager. Upon receiving the travel plan from the intersection manager, the vehicle will conduct a block verification (as elaborated in Section IV-B1) to verify the travel plan. If so, the vehicle will follow the plan until it leaves the intersection if there are not any threats. Note that in case the vehicle needs to change its destination, it can send the change request to the intersection manager to obtain a new travel plan.
- **Self-evaluation**: If the travel plan verification fails due to an invalid block or erroneous travel plans contained in the block, the vehicle will deem the intersection manager has been compromised and enter the self-evacuation mode to find a safe route to leave the intersection. Upon leaving, the vehicle will also broadcast the problem (denoted as a global report) to the vehicles at the intersection. Other vehicles will react to such reports following the protocols in the global verification stage as described below.
- **Global Verification**: If a vehicle receives multiple global reports from peer vehicles claiming that the intersection manager may be under attack, the vehicle will enter the global verification stage. The vehicle will collect blocks of travel plans from vehicles at the intersection to check the consistency (Section IV-B3). If the travel plans are incorrect, the vehicle will enter the self-evacuation mode.
- **Local Verification**: A vehicle also has a role of local verifier as long as there are other vehicles around it. This is essentially the idea of the neighborhood watch. Peer vehicles supervise neighboring vehicles and will report any abnormal behavior immediately to the intersection manager (detailed algorithms are in Section IV-B2). If at the time of incident reporting, the intersection manager is still functioning (i.e., endowed by global verifiers), the reporting vehicle will wait for the intersection manager to dismiss the alarm or generate evacuation plans. If there is no response from the intersection manager, the reporting vehicle will enter the self-evacuation mode before the evacuation time runs out and also send out global reports to warn other vehicles regarding the potential threats.

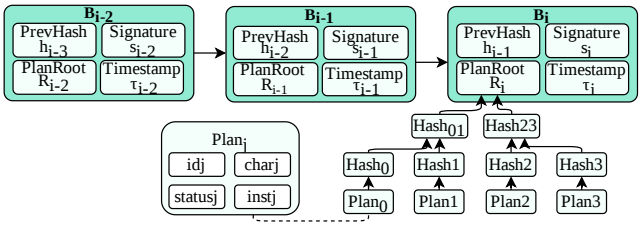

Fig. 3. The Data Structure of the Travel Plan Blockchain

B. Algorithms of the NWADE Mechanism

In what follows, we elaborate the detailed algorithms at each state of the NWADE mechanism.

1) **Block Packaging and Verification:** As vehicles are streaming into intersections continuously, the intersection manager needs to keep generating travel plans for incoming vehicles. For the integrity of the travel plans, i.e., to prevent the compromised intersection manager from sending conflicting travel schedules to different vehicles to cause the collision, all the travel plans are stored as a blockchain as follows. Note that each vehicle only needs to store the blockchain at its current intersection. It can delete the blockchain after it passes the intersection so there is not much storage overhead.

For vehicles coming during the same processing window, the intersection manager will generate the travel plans for this batch of vehicles and store them in one block $\mathcal{B}_i$:

$$\mathcal{B}_i = \langle s_i, h_{i-1}, \tau_i, \mathcal{R}_i \rangle \tag{1}$$

In block $\mathcal{B}_i$, $s_i = Sign(\langle h_{i-1}, \tau_i, \mathcal{R}_i \rangle, K_r)$ is the signature of the block generated by the intersection manager using its private key K_r; h_{i-1} is the hash value of the previous block $\mathcal{B}_{i-1}$ generate during the prior processing window; τ_i is the current timestamp; and $\mathcal{R}_i$ is the root of a hash-value tree that contains all the newly generated travel plans at the leave nodes and the hash values of the travel plans as internal nodes. Each travel plan $\mathcal{T}_i^j$ for vehicle V_j is in the form of $\langle id_j, char_j, status_j, inst_j \rangle$, where id_j is the identity of the vehicle, $char_j$ is V_j's static characteristics, $status_j$ is V_j's dynamic status, and $inst_j$ is the detailed instruction for the vehicle to follow to cross the intersection. According to the specific design of the intelligent intersection management system, the id_j could be an anonymous identity to protect privacy; the static characteristics of a vehicle could be car brand, model, and color; the dynamic status may include the vehicle's GPS coordinates, speed, and moving direction. Fig. 3 gives an overview of the data structure of the travel plan blockchain.

The newly generated block will be broadcast to all the vehicles at the intersection. Each vehicle will then execute the following verification protocol (Algorithm 1):

(i) When receiving a block B_i, vehicle V_x validates the signature s_i using the intersection manager's public key K_u to check if the block B_i is issued by the intersection manager. If the verification fails, it is likely the intersection manager has been compromised and vehicle V_x will enter self-evacuation mode and send out a global report to warn other vehicles.

(ii) If the new block's signature is correct, vehicle V_x will further calculate the travel plans in the block to see if the plans contain any conflict (i.e., car collision). If the plans are conflicting, it is again likely the intersection manager has been attacked and vehicle V_x will start self-evacuation.

(iii) After the plan validation, vehicle V_x will store the block. If vehicle V_x is a new vehicle, the verification process stops here. If vehicle V_i entered the intersection earlier and has received other blocks, it will further verify whether the new block is part of the blockchain as follows. Let B_{i-1} denote the last block in the chain, V_x will check if $hash(\mathcal{B}_{i-1}) = h_{i-1}$ where h_{i-1} is the hash value in the new block B_i. If not, V_x will again enter the self-evacuation mode.

(iv) If V_x holds multiple blocks, it will further check if the travel plans in the new block have any conflict with the previous plans it received. If not, the verification process completes. The maximum length of the chain that a vehicle needs to cache and verify equals τ/δ, which is the time (τ) that a vehicle needs to cross the intersection divided by the time interval δ that the intersection manager processes a batch of vehicles. Considering the physical capacity of an intersection and the short period of crossing time, the number of blocks to be stored and verified should be within the computational capability of each vehicle.

Algorithm 1: Block Verification

1. $B_i =< s_i, h_{i-1}, \tau_i, \mathcal{R}_i > \xleftarrow{broadcast} IMU$
2. if(!$ValidateSign(\langle h_{i-1}, \tau_i, \mathcal{R}_i \rangle, s_i, K_u)$):
3. Goto line 12
4. else if($HasConflict(B_i)$):
5. Goto line 12
6. else if vehicle has previous blocks:
7. if ($hash(B_{i-1})! = B_i.h_{i-1}$):
8. Goto line 12
9. else if B_i has conflicts with $B_{i-k}..B_{i-1}$:
10. Goto line 12
11. Block verified, store B_i and return
12. Self-evacuation and send out global report

It is worth noting that the chance of entering the self-evacuation mode is very low as calculated in Section IV-B4 under the practical assumption that the majority of vehicles are normal vehicles. The use of the blockchain for storing the travel plans guarantees the integrity and consistency of the travel scheduling for all the vehicles at the intersection. This process ensures that no one can modify or counterfeit the travel plan without being noticed. Moreover, in case of packet loss, a vehicle can request the blocks from neighboring vehicles or from the intersection manager without worrying that the block

might be altered.

2) **Local Verification and Report Verification:** In the NWADE mechanism, individual vehicles have an important task which is conducting a "neighborhood watch". As autonomous vehicles are typically equipped with various types of devices and sensors such as cameras, LiDAR, and radar sensors for detecting traffic and road conditions, these sensing abilities are sufficient to monitor neighboring vehicles' behaviors. We leverage such sensing abilities to conduct so-called local verification. The goal of local verification is to detect suspicious vehicles as early as possible to prevent car accidents. The specific local verification protocol is as follows (Algorithm 2):

(i) First, vehicle V_x checks if it has travel plans for the neighboring vehicles' requests by matching them with vehicles' descriptions (e.g., car brand, model, color, speed, location) including the travel plans. Since nearby vehicles are mostly entering the intersection during the same time interval, the travel plans of neighboring vehicles are likely to be in the same block of vehicle V_x's own travel plan.

(ii) In the case that some vehicles came in an earlier time interval, vehicle V_x will request the blocks from those vehicles in front of it. The received blocks will go through block verification whereby the block signature and hash values will be checked as discussed in the previous block verification protocol. This ensures that the received blocks are legitimate.

(iii) After obtaining the corresponding travel plans for neighboring vehicles, vehicle V_x will calculate the expected status (i.e., location and speed) of its neighboring vehicles (denoted as V_y) and compare the calculated status with the detected status of V_y. If the difference from the travel plan is larger than a tolerance threshold, that means V_y is deviating from its assigned travel plan and is likely under attack. In that case, V_x will report this abnormality to the intersection manager by sending the incident report in the form of $\mathcal{IR} = \langle \mathcal{E}_\dagger, \mathcal{B}_y \rangle$, where $\mathcal{E}_\dagger$ is the evidence of the current status of vehicle V_y, i.e., the related data of the on-board sensors in V_x, and the block $\mathcal{B}_y$ that contains V_y's travel plan.

Once received an incident report from vehicle V_x regarding a suspicious vehicle V_y, the intersection manager will start the report verification process as follows.

(i) If the intersection manager has the ability to detect the status of vehicles at the intersection such as using the camera, it will directly check if vehicle V_y follows the designated travel plan. If V_y is malicious, the intersection manager will enter the evacuation mode and broadcast evacuation plans for other vehicles.

(ii) If the intersection manager has limited detection capabilities, it will ask vehicles around V_y to conduct local verification. If the majority of the returned reports indicate that V_y is abnormal, it will first enter the evacuation mode for safety concerns. Meanwhile, it will request the local verification from another group of vehicles to double-check the status of V_y. This is to prevent a group of malicious vehicles from using false reports about an actual normal vehicle V_y to slow down the traffic. By using different groups of vehicles for verification, the chance of having malicious vehicles dominating the majority of the vehicles on the road segment becomes very low. Let P_d denote the probability for the intersection manager to identify such kind of attack is very high. The relationship between P_d and the number of malicious vehicles can be estimated as shown in Equation 2, where k denotes the number of vehicles being compromised, p_v denotes the probability for the attacker to compromise a vehicle, and function e and parameter ω is to regularize the probability value. P_d is inversely proportional to the number of malicious vehicles on the same road segment. Although the detection difficulty increases with the increase of the number of malicious vehicles, we should also note that the probability of successfully controlling the larger number of vehicles decreases even faster. Therefore, the overall probability for the intersection manager to identify such kind of attack is very high.

$$P_d = \frac{1}{e^{\omega \cdot k \cdot (p_v)^k}} \tag{2}$$

(iii) After verification, if vehicle V_y is confirmed to be normal, the intersection manager will inform the reporting vehicle V_x to dismiss the alarm. Also, the intersection manager will record V_x's identity for future reference in case V_x is malicious and repeatedly sends out false alarms. If V_x did not receive the confirmation or the evacuation plan from the IMU, it will assume that the IMU has been compromised and refuse to validate the report. V_x will then enter self-evacuation mode and send out global reports.

3) **Global Verification:** The global verification protocol is used to handle the situation in case the intersection manager may be under the control of an attacker. If a vehicle V_x receives global reports broadcasted by other peer vehicles, vehicle V_x

Algorithm 2: Local Verification

1. if V_x has neighboring vehicle V_y's plan
2. $\quad p_y \leftarrow$ obtain neighboring vehicles' plan
3. else:
4. $\quad B_y \leftarrow$ block from the vehicles in front of V_x
5. $\quad p_y \leftarrow$ obtain V_y's plan from B_y
6. $status_{y\_expected} \leftarrow$ calculated based on p_y
7. $status_{y\_detected} \leftarrow$ detect V_y's current status
8. $diff \leftarrow Diff(status_{y\_expected}, status_{y\_detected})$
9. if $diff$ larger than tolerance threshold:
10. $\quad$ Report $\xrightarrow{\mathcal{IR}=\langle \mathcal{E}_\dagger, \mathcal{B}_y \rangle} IMU$
11. Wait for the IMU's response
12. if IMU refuses to response:
13. $\quad$ Self-evacuation and send out global report

Algorithm 3: Global Verification

1. Receive $reports_{global}$ from other vehicles
2. if $reports_{global}$ reports conflicting travel plans:
3. Obtain B_e
4. if B_e contains conflict plans:
5. Self-evacuation and send out global report
6. else if $reports_{global}$ reports abnormal vehicles V_y:
7. if V_y is nearby:
8. V_x will perform local verification
9. else:
10. V_x will analyze V_y's travel path
11. if # of global reports exceed threshold:
12. Enter self-evacuation mode

will execute the global verification protocol as follows to make the decision (Algorithm 3).

(i) ***Conflicting travel plans***. If vehicle V_x receives multiple global reports claiming that some of the blocks (denoted B_e) sent by the intersection manager contain conflicting travel plans, vehicle V_x will first check if it has received the same block. If not, that means B_e must be generated before its entrance to the intersection. Thus, it will request B_e from vehicles in front of it. Since blocks are broadcasted to every vehicle, if the same block has passed its own verification, it can easily conclude that the received global reports are malicious and will send an incident report to the intersection manager. If B_e does contain conflicting travel plans with the travel plans in the latest blocks, vehicle V_x will consider the intersection manager has been compromised and will enter self-evacuation mode and send out the global report to warn other vehicles as well. Given the assumption that the majority of the vehicles are benign, as more vehicles send out global reports, it will become easier for later vehicles to conclude that the intersection manager is no longer trustworthy.

(ii) ***Abnormal vehicles***. If vehicle V_x receives global reports claiming the existence of a malicious vehicle V_y and ignorance of the intersection manager, vehicle V_x will first check if V_y is nearby. If so, vehicle V_x will perform its own local verification. If not, vehicle V_x will analyze V_y's travel path based on the global report and its own travel route. If V_x is far away from V_y and has sufficient time to evacuate, V_x will enter the self-evacuation mode only if the number of the global reports with respect to V_y exceeds a safety threshold (as discussed in the next subsection). This is because more and more honest vehicles will detect and report the misbehavior of V_y as time passes no matter whether the intersection manager is responsive or not.

4) **Self-Evacuation:** As discussed in the previous protocols, there are multiple situations when a vehicle needs to self evacuate because the intersection manager is no longer trustworthy. These situations include the failure of block verification, failure of receiving the response from the intersection manager, and the receipt of a large number of global reports. Once a vehicle enters the self-evacuation mode, it depends on the individual vehicles' onboard system to either pull over to the roadside or finds the safest route to exit the intersection as quickly as possible. Here, we would like to stress that the probability of entering the self-evacuation mode is actually very low given that it is extremely hard to compromise the intersection manager and a large number of vehicles at the intersection. Specifically, the self-evacuation probability can be estimated as follows.

Let p_{im} denote the probability that an intersection manager is compromised, p_v denotes that an individual vehicle is compromised, and let p_{loc} denote the probability that the compromised vehicle is near the location loc. Without loss generality, we can assume that $p_{im} << p_v$ as the intersection manager is supposed to be much better protected than individual vehicles. The probability that a vehicle needs to self-evacuate (denoted as P_e) can be estimated as shown in Equation 3.

$$P_e = 1 - (1 - p_{im})(1 - (p_v p_{loc})^k) \tag{3}$$

In Equation 3, $(p_v p_{loc})^k$ is the probability when k vehicles have been compromised and are gathering near the location loc. This probability quickly becomes smaller when k increases. From the attacker's perspective, the more vehicles it attacks, the easier it crashes the traffic management system. However, the likelihood of simultaneously taking control of a large number of vehicles decreases fast. Therefore, we use $(1 - p_{im})(1 - (p_v p_{loc})^k)$ to estimate the probability when there is no attack, i.e., no need to evacuate. By subtracting this probability from 1, we obtain the probability when a vehicle needs to self evacuate. We now plug in some specific numbers to have a better understanding of how small this evacuation probability would be. For example, assume that $p_v p_{loc}$ equals 10% and p_{im} equals 0.1%, and the number of vehicles within a vehicle's sensing and communication range is around 20 (medium density). If the attacker tries to control k vehicles near a location to trigger the self-evacuation phase, the value of k should be larger than half of the number of vehicles around the location in order to win the majority vote, which would be 20/2+1=11 vehicles in this case. By plugging the numbers to Equation 3, we obtain the self-evacuation probability P_e=1-(1-0.001)·(1-0.1^{11})≈0.1%. The safety threshold for a vehicle that is far away from the suspicious vehicle can be set accordingly to reduce the false alarm rate.

5) **Evacuation and Post-evacuation Recovery:** When the intersection manager is trustworthy and detected malicious vehicles, the intersection manager will start the evacuation process to protect normal vehicles. First, the intersection manager will send out an alert message to all vehicles that contain the suspect vehicle's identifiable features (e.g., car model, brand, color) and location. Meanwhile, the intersection manager will generate and broadcast new travel plans by considering the location and moving status of the malicious vehicles to help

normal vehicles circumvent them. It is worth noting that the evacuation plans can be generated very quickly and will be instantly available for the needed vehicles. The generation process of the evacuation plan is similar to that of the initial travel plans. For example, it takes the IMU less than 0.5 seconds to generate the travel plans for a 4-way intersection with 1000 vehicles as reported in [16]. For the vehicles that are certain distance away from the malicious vehicle, they will have time to follow the evacuation plans to avoid encountering the malicious vehicle. For the vehicles which are very close to the malicious vehicle, they should have already detected the malicious vehicle through their own sensors and started self-evacuation. If there are newly identified malicious vehicles during the evacuation, the detection scheme is the same as in the pre-evacuation stage and the intersection manager will regenerate the travel plans for normal vehicles based on the latest status. The evacuation plans will also be packaged in the blockchain just like regular travel plans to ensure integrity.

After the safety threats are cleared such as that the malicious vehicle left the intersection, the intersection manager will enter the post-evacuation recovery phase, which is essentially preparing to generate normal travel plans. This is because evacuation plans may instruct vehicles to drive slower to maintain sufficient reaction to any sudden movement change of the malicious vehicles. During the post-evacuation phase, the intersection manager will gradually bring the vehicles to normal and fast passing speed. Again, the specific scheduling will be determined by the actual traffic scheduling system as our NWADE is focused on providing safety protocols.

V. Security Analysis

Let us revisit the security questions raised in the introduction:

Can the incoming vehicles trust the travel plans generated by the intersection manager? How can they know the plan would not cause collisions? With our NWADE mechanism in place, incoming vehicles can now verify the integrity of the received travel plans and also calculate the travel plans of other vehicles in the received blocks to ensure the correctness of the travel plans.

Can the intersection manager trust the vehicle which reports spotting a malicious vehicle nearby? The intersection manager can judge the trustworthiness of the incident report through the help of other local verifiers. As we just discussed, even if a group of malicious vehicles attempts to game the majority voting on one leg of the intersection, there is still a high probability for the intersection manager to identify the wrong reports promptly.

Will a majority vote of vehicles surrounding a suspicious vehicle be sufficient to confirm or clear the alarm? What if there are a group of malicious vehicles traveling together to game the majority vote scheme? This issue can be resolved by our NWADE mechanism as presented in Section IV-B2.

Can the vehicles trust the evacuation plan broadcasted by the intersection manager or other peer vehicles? What if that is just a sham because the intersection manager or the peer vehicle has been compromised? First, the travel plans in the evacuation broadcast can be verified in a similar way as regular travel plans to prevent the intersection manager from using conflicting travel plans to induce collisions. The evacuation warnings (i.e., the global reports in the NWADE mechanism) sent by peer vehicles can be validated via simple majority voting under the assumption that the majority of vehicles at the intersection are benign. This is because individual vehicles will all send out global reports once they enter the self-evacuation mode. The misleading reports sent by a small group of malicious vehicles will not be able to dominate the entire intersection. The malicious parties can at most slow down the traffic for a short period.

VI. Experimental Study

A. Experimental Settings

The proposed NWADE mechanism is implemented using ECMA Script 2015 and integrated into the most recent intelligent intersection management system DASH [16] as it can handle various shapes of intersections. It is worth noting that our mechanism can be integrated to other traffic management systems as well. For evaluation, we developed a 3D intelligent intersection traffic simulator for large-scale evaluation. All the experiments are run in macOS 10.15 with a 3.2 GHz Intel i7 CPU and 16GB memory.

In the experiments, we evaluated five popular types of intersections: (i) 3-way roundabout; (ii) 4-way cross; (iii) 5-way irregular intersection; (iv) 4-way continuous flow intersection (CFI); and (v) 4-way diverging diamond interchange (DDI). We tested these types of intersections under eleven attack schemes to demonstrate the effectiveness of our mechanism. The traffic flow is generated by a Poisson distribution with the vehicle density ranging from 20 to 120 vehicles per minute. By changing the vehicle density, we can evaluate the performance of the NWADE mechanism in the intersections with different capacities and road conditions. A density of 20 vehicles per minute simulates a small town scenario which has a longer distance between two vehicles and a higher average moving speed. Take a 4-way intersection as an example, a density of 20 vehicles per minute means only 5 vehicles per minute for each lane on average. When the density reaches 120 vehicles per minute, it simulates scenarios in big cities whereby vehicles are more crowded and the intersection may reach its capacity. If not otherwise specified, we choose 80 vehicles per minute as the default setting. Based on the real-world statistics, we set the percentages of left-turn, going-straight, and right-turn vehicles as 25%, 50%, and 25%, respectively, and set the default speed limit as 50 mph (80 km/h), max acceleration as 6.6 ft/s^2 (2 m/s^2), and max deceleration as 10.0 ft/s^2 (3 m/s^2). We set the maximum communication radius as 1500 ft (457 m), and the network latency as 30 milliseconds. We vary the sensing radius of the vehicles to obtain the surrounding vehicle's status from 300 ft (91 m) to 1000 ft (305 m). Since we are targeting future applications, if not otherwise specified, we choose 1000 ft (305 m) as the default setting for the perception range of both the vehicles and the intersection

TABLE I
ATTACK SETTINGS

Attack Setting	Number of malicious vehicles	Intersection manager	Plan violations	False reports
V1	1	Benign	1	0
V2	2	Benign	1	1
V3	3	Benign	1	2
V5	5	Benign	1	4
V10	10	Benign	1	9
IM	0	Malicious	0	0
IM_V1	1	Malicious	1	0
IM_V2	2	Malicious	1	1
IM_V3	3	Malicious	1	2
IM_V5	5	Malicious	1	4
IM_V10	10	Malicious	1	9

TABLE II
FALSE ALARM RATE

Attack Setting	False Alarm Type A (Trigger/Detection Rate)	False Alarm Type B (Trigger/Detection Rate)
V1, V2, V3, V5	0% / 100%	0% / 100%
V10	5% / 100%	0% / 100%
IM	0% / 100%	N/A
IM_V1	0% / 100%	N/A
IM_V2	0% / 100%	N/A
IM_V3	0% / 100%	N/A
IM_V5	9% / 100%	N/A
IM_V10	14% / 100%	N/A

manager. This detection range is already achieved by the existing LiDAR systems [2]. The hash value of a block is generated using the SHA256 method and the length of the intersection manager's private key $\mathcal{K}_r$ is 2048 bits.

In the experiments, we simulate the attack settings as shown in Table I. These settings are corresponding to the 4 threat models as presented in Section III, whereby we simulate the existence of a single malicious vehicle ($V1$), multiple malicious vehicles ($V2$ to $V10$), a malicious intersection manager (IM), and the collusion between the intersection manager and vehicles (IM_V1 to IM_V10). For each setting, we test 10 rounds. In each round, we randomly choose the positions and vehicles to perform the attacks.

B. Effectiveness Evaluation

In the first round of experiments, we aim to evaluate if our proposed NWADE mechanism can successfully identify and validate attacks in varied vehicle densities and attack settings. We first measure the detection rate of false alarms and then report the detection rate of real incidents.

There are two main types of false alarms: (i) False Alarm Type A where the attacker(s) sends a false claim that there is a vehicle violating the travel plan; (ii) False Alarm Type B where the attacker(s) send a false claim that the intersection manager is sending wrong travel plans to cause a collision.

As presented in Table. II, there is a slim chance for attackers to trigger the self-evacuation using the false claim type A. This is because in most cases, there are a sufficient number of benign vehicles to conduct the correct verification and help

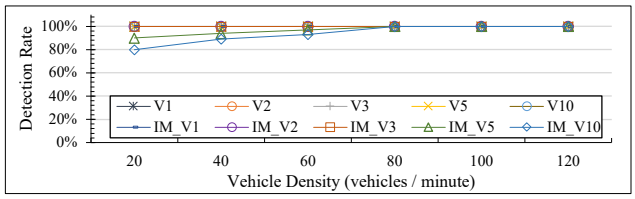

Fig. 4. Detection Rate under Different Vehicle Densities

the intersection manager dismiss the wrong claim. In the case that the intersection manager is also compromised and sends out evacuation plans to vehicles directly, such misbehavior will first be detected by vehicles near the wronged vehicle since benign vehicles conduct local verifications continuously. Benign vehicles will then send out global reports to warn other vehicles that the intersection manager is no longer trustworthy. When the benign vehicles are outnumbered by the malicious vehicles at the same scene such as in the setting IM_V10, it is harder for other peer vehicles to decide whether the global reports are trustworthy and they may enter the self-evacuation mode for the safety caution. Regarding false alarm type B, all the attack attempts will fail no matter whether the wrong travel plans were sent by malicious vehicles or the intersection manager. This is because vehicles can simply verify the blockchain and validate the correctness of the travel plans.

Next, we test if the NWADE mechanism can successfully validate the malicious vehicles' misbehavior reported by benign vehicles. We vary the density from 20 to 120 vehicles per minute in a common 4-way cross with 10 incoming lanes. As shown in Fig. 4, when a single vehicle or multiple vehicles have been compromised, the travel plan violation can be 100% detected no matter whether the intersection manager is benign or not. This is because as vehicles keep moving, their neighbors are changing over time, which means local verification will be conducted by different vehicles especially when the malicious vehicles are spread at different legs of the intersection. Even if the malicious vehicles move as a group, the probability that the attacker dominates the majority of the vehicles near the same incident spot for a period of time is still very low. The most challenging case is when the intersection manager is colluding with a group of malicious vehicles (settings IM_V1 *to* IM_V10), whereby the normal vehicles still have more than 80% chance to detect such a problem. Note that vehicles near the incident spot can always enter the self-evacuation mode without waiting for global consensus.

C. Efficiency Evaluation

Fig. 5 shows the time taken to detect the reports of malicious vehicles' deviating from travel plans and the reports of malicious vehicles sending wrong travel plans at a 4-way intersection. Observe that the detection time for both cases is less than 360 milliseconds. Assuming that the malicious vehicle moves at a speed of 50 mph (80 km/h), the maximum

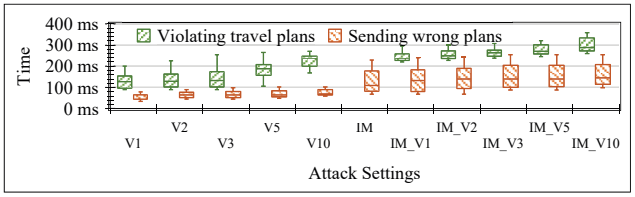

Fig. 5. Detection Time

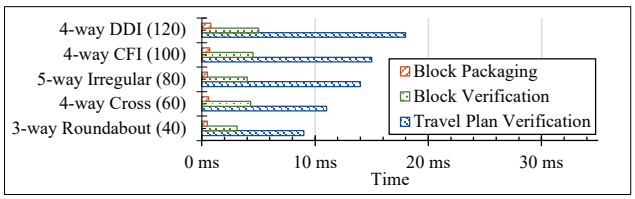

Fig. 6. Block Chain Management and Verification

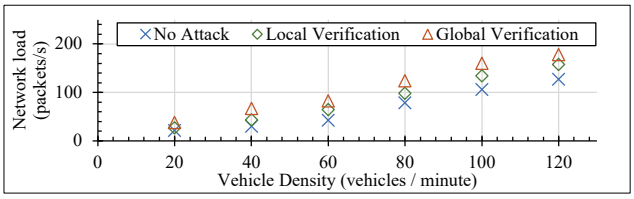

Fig. 7. Network Load

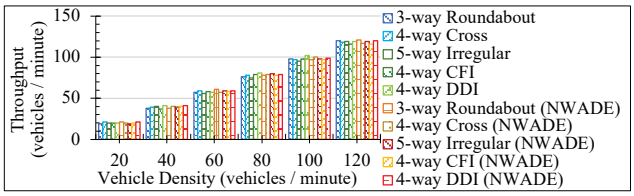

Fig. 8. Traffic Throughput

displacement of the vehicle will be 26.2 ft (8.0 meters). This leaves enough room for other vehicles at the intersection to avoid the danger. It is worth noting that, for the normal vehicles, if they have witnessed the nearby malicious vehicle, they can take actions immediately to avoid accidents without waiting for others' verification results.

We now examine the time needed for blockchain management and verification at both the vehicle side and the intersection manager side. In Fig. 6, the y-axis lists the types of intersections and the vehicle densities being tested. For example, 4-way DDI (120) refers to the 4-way diverging diamond interchange with 120 vehicles per minute. We can observe that the overall calculation time is less than 20 milliseconds. Assuming that a vehicle moves at a speed of 50 mph (80 km/h), the displacement of the vehicle will be less than 1.5 feet (0.45 meter), which will not affect the vehicles to make timely decisions to avoid possible collisions.

We further evaluate the overall network load introduced by the proposed NWADE mechanism. Fig. 7 shows the total number of packets in the network in a 4-way intersection as shown in Fig. 1 under three types of events: (i) no attack; (ii) local reports sent; (iii) global reports sent. We can see that this experiment result shows that the amount of the packets needed by the NWADE mechanism is reasonably small and the mechanism would be practical in the real world.

Finally, we study the overhead of NWADE on the overall traffic efficiency when there is no attack. We compare the traffic throughput at five different intersections with and without the NWADE mechanism. Fig. 8 shows that the throughput at the intersection stays almost the same after adding the NWADE mechanism regardless of the types of the intersections and the vehicle density.

VII. Conclusion

In this paper, we propose a sophisticated security mechanism, NWADE, to assist the intelligent intersection management system to provide strong security guarantees to autonomous vehicles. It is the first time that complicated and challenging threat scenarios during automatic traffic scheduling are systematically analyzed and tackled. The proposed NWADE mechanism leverages blockchain technology and the collaborative neighborhood watching concept to ensure the safety of vehicles under a variety of attacks. The NWADE mechanism is robust even if the intersection manager and multiple vehicles have been compromised. We have integrated the NWADE mechanism into the latest intersection management system and tested various types of intersections and traffic flows in our developed 3D traffic simulation platform. The experimental results demonstrate that our approach is not only very effective in mitigating security threats, but also introduces negligible computation overhead. In the future, we are interested in exploring the more challenging scenario during the transitional period when there is a mix of autonomous vehicles and legacy vehicles.

References

[1] T. Abera, R. Bahmani, F. Brasser, A. Ibrahim, A.-R. Sadeghi, and M. Schunter, "Diat: Data integrity attestation for resilient collaboration of autonomous systems." in *NDSS*, 2019.

[2] Aurora, "The power of fmcw lidar + scale: Why acquiring ours lidar unlocks the commercialization of the aurora driver," www.aurora.tech, 2021, https://aurora.tech/blog/the-power-of-fmcw-lidar-and-scale-acquiring.

[3] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, G. Fortino, and M. Villari, "A multi-agent autonomous intersection management (ma-aim) system for smart cities leveraging edge-of-things and blockchain," *Information Sciences*, vol. 522, pp. 148–163, 2020.

[4] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, and M. Villari, "A secure and dependable multi-agent autonomous intersection management (ma-aim) system leveraging blockchain facilities," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. IEEE, 2018, pp. 226–231.

[5] A. Buzachis, B. Filocamo, M. Fazio, J. A. Ruiz, M. Á. Sotelo, and M. Villari, "Distributed priority based management of road intersections using blockchain," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1159–1164.

[6] Q. A. Chen, Y. Yin, Y. Feng, Z. Mao, and H. Liu, "Exposing congestion attack on emerging connected vehicle based traffic signal control," in *NDSS*, 01 2018.

[7] A. P. Chouhan and G. Banda, "Autonomous intersection management: A heuristic approach," *IEEE Access*, vol. 6, pp. 53 287–53 295, 2018.

[8] S. Crow, "You'll spend this much of your life waiting at red lights," BestLifeOnline.com, 2018, https://bestlifeonline.com/red-lights/.

[9] H.-W. Ferng, J.-Y. Chen, M. Lotfolahi, Y.-T. Tseng, and S.-Y. Zhang, "Messages classification and dynamic batch verification scheme for vanets," *IEEE Transactions on Mobile Computing*, 2019.

[10] Y. Guo, J. Ma, C. Xiong, X. Li, F. Zhou, and W. Hao, "Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach," *Transportation research part C: emerging technologies*, vol. 98, pp. 54–72, 2019.

[11] S. Gurung, D. Lin, A. Squicciarini, and E. Bertino, "Information-oriented trustworthiness evaluation in vehicular ad-hoc networks," in *International conference on network and system security*. Springer, 2013, pp. 94–108.

[12] S. Gurung, A. Squicciarini, D. Lin, and O. K. Tonguz, "A moving zone based architecture for message dissemination in vanets," in *8th international conference on network and service management and workshop on systems virtualiztion management*. IEEE, 2012, pp. 184–188.

[13] M. G. M. M. Hasan, A. Datta, M. A. Rahman, and H. Shahriar, "Chained of things: A secure and dependable design of autonomous vehicle services," in *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2018, pp. 498–503.

[14] W. Jiang, F. Li, D. Lin, and E. Bertino, "No one can track you: randomized authentication in vehicular ad-hoc networks," in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 197–206.

[15] J. Kang, Y. Elmehdwi, and D. Lin, "Slim: Secure and lightweight identity management in vanets with minimum infrastructure reliance," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2017, pp. 823–837.

[16] J. Kang and D. Lin, "Dash: A universal intersection traffic management system for autonomous vehicles," in *IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 89–99.

[17] J. Kang, D. Lin, E. Bertino, and O. Tonguz, "From autonomous vehicles to vehicular clouds: Challenges of management, security and dependability," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1730–1741.

[18] J. Kang, D. Lin, W. Jiang, and E. Bertino, "Highly efficient randomized authentication in vanets," *Pervasive and Mobile Computing*, vol. 44, pp. 31–44, 2018.

[19] S. Karumanchi, A. Squicciarini, and D. Lin, "Selective and confidential message exchange in vehicular ad hoc networks," in *International Conference on Network and System Security*. Springer, 2012, pp. 445–461.

[20] ——, "Privacy-aware access control for message exchange in vehicular ad hoc networks," *Telecommunication Systems*, vol. 58, no. 4, pp. 349–361, 2015.

[21] M. Khayatian, M. Mehrabian, E. Andert, R. Dedinsky, S. Choudhary, Y. Lou, and A. Shirvastava, "A survey on intersection management of connected autonomous vehicles," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, pp. 1–27, 2020.

[22] Q. Kong, R. Lu, H. Zhu, and M. Ma, "Achieving secure and privacy-preserving incentive in vehicular cloud advertisement dissemination," *IEEE Access*, vol. 6, pp. 25 040–25 050, 2018.

[23] P. Kremer, I. Koley, S. Dey, and S. Park, "State estimation for attack detection in vehicle platoon using vanet and controller model," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[24] C. Lai, G. Li, and D. Zheng, "Spsc: A secure and privacy-preserving autonomous platoon setup and communication scheme," *Transactions on Emerging Telecommunications Technologies*, p. e3982, 2020.

[25] Z. Li, Q. Wu, H. Yu, C. Chen, G. Zhang, Z. Z. Tian, and P. D. Prevedouros, "Temporal-spatial dimension extension-based intersection control formulation for connected and autonomous vehicle systems," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 234–248, 2019.

[26] J. Liang and M. Ma, "Ecf-mrs: An efficient and collaborative framework with markov-based reputation scheme for idss in vehicular networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 278–290, 2020.

[27] D. Lin, J. Kang, A. Squicciarini, Y. Wu, S. Gurung, and O. Tonguz, "Mozo: a moving zone based routing protocol using pure v2v communication in vanets," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1357–1370, 2017.

[28] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, 2018.

[29] W. Luo and W. Ma, "Efficient and secure access control scheme in the standard model for vehicular cloud computing," *IEEE Access*, vol. 6, pp. 40 420–40 428, 2018.

[30] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transportation research part C: emerging technologies*, vol. 100, pp. 161–176, 2019.

[31] S. M. Pournaghi, B. Zahednejad, M. Bayat, and Y. Farjami, "Necppa: A novel and efficient conditional privacy-preserving authentication scheme for vanet," *Computer Networks*, vol. 134, pp. 78–92, 2018.

[32] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, "A blockchain framework for securing connected and autonomous vehicles," *Sensors*, vol. 19, no. 14, p. 3165, 2019.

[33] A. Squicciarini, D. Lin, and A. Mancarella, "Paim: Peer-based automobile identity management in vehicular ad-hoc network," in *2011 IEEE 35th Annual Computer Software and Applications Conference*. IEEE, 2011, pp. 263–272.

[34] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 779–811, 2018.

[35] Y. Wu, H. Chen, and F. Zhu, "Dcl-aim: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 246–260, 2019.

[36] Y. Xia, W. Chen, X. Liu, L. Zhang, X. Li, and Y. Xiang, "Adaptive multimedia data forwarding for privacy preservation in vehicular ad-hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2629–2641, 2017.

[37] B. Xu, S. E. Li, Y. Bian, S. Li, X. J. Ban, J. Wang, and K. Li, "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 322–334, 2018.

[38] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[39] C. Yu, Y. Feng, H. X. Liu, W. Ma, and X. Yang, "Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections," *Transportation Research Part B: Methodological*, vol. 112, pp. 89–112, 2018.

[40] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "Metalight: Value-based meta-reinforcement learning for traffic signal control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1153–1160.

[41] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "Smaka: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Transactions on Information Forensics and Security*, 2020.

[42] L. Zhang, "Otibaagka: A new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2998–3010, 2017.

[43] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 516–526, 2017.