TriP: Misbehavior Detection for Dynamic Platoons using Trust

Garlichs, Keno and Willecke, Alexander and Wegner, Martin and Wolf, Lars

Authors post-print published on 2020-07-22
Originally published in *IEEE Intelligent Transportation Systems Conference (ITSC)*
Publisher version available at https://ieeexplore.ieee.org/abstract/document/8917188/
DOI: 10.1109/ITSC.2019.8917188

Abstract:

Platooning is able to improve fuel efficiency andreduce road congestion. But to maximize the concept's impact,platoons need to be created dynamically whenever feasible.Therefore, vehicles have to cooperate with unknown and pos-sibly malicious partners, creating new safety hazards. Hence,vehicles need to be able to determine the trustworthiness of theircooperators. This paper proposes TriP, a trust model whichrates platoon members by the divergence of their reportedto their actual behavior. The proposed model is evaluatedagainst attacks from literature. The evaluation demonstratesthat TriP detects all attacks and prevents harm by deployingcountermeasures thus mitigating safety hazards.

TriP: Misbehavior Detection for Dynamic Platoons using Trust

Keno Garlichs[1], Alexander Willecke[1], Martin Wegner[1] and Lars C. Wolf[1]

Abstract— **Platooning is able to improve fuel efficiency and reduce road congestion. But to maximize the concept's impact, platoons need to be created dynamically whenever feasible. Therefore, vehicles have to cooperate with unknown and possibly malicious partners, creating new safety hazards. Hence, vehicles need to be able to determine the trustworthiness of their cooperators. This paper proposes TriP, a trust model which rates platoon members by the divergence of their reported to their actual behavior. The proposed model is evaluated against attacks from literature. The evaluation demonstrates that TriP detects all attacks and prevents harm by deploying countermeasures thus mitigating safety hazards.**

I. INTRODUCTION

Recent advances in Intelligent Transport Systems (ITS) brought a multitude of Advanced Driver Assistance Systems (ADAS), which make use of a plethora of sensors, increased processing power and Vehicle-to-Everything (V2X) communication. Equipped vehicles are not only capable of sensing their surroundings and warning their drivers and other road users. Even automated driving functions are possible.

In a platoon for example, a chain of vehicles driving towards the same destination, while the members keep a constant inter-vehicle spacing, beacon messages containing the vehicle's position, velocity and acceleration are used to increase stability of the cruise controllers by a priori informing other members about planned maneuvers. Therefore, vehicles can reduce safety distances to each other. This can have a large positive effect on road usage efficiency and especially on fuel economy while also increasing road safety [1]. With existing platooning controllers like Cooperative Adaptive Cruise Control (CACC) [2] , the platoon leader controls the platoon by announcing its own kinematic data, the platoon's target velocity and inter-vehicle gap, which are used by succeeding vehicles to follow the leader autonomously [3], [4]. Beacons of other platoon members are incorporated to control acceleration and maintain a constant inter-vehicle spacing.

Platooning demonstrators like PATH [5] and SARTRE [6] focus on the practical implementation to demonstrate the concept's feasibility, and therefore platoons consist of an invariable and small set of member vehicles. Due to the small and predefined set of vehicles used in those experiments, an ultimate trust relationship between the vehicles can be implied and thus the exchanged kinematic data was always deemed trustworthy.

Predefined groups and a limited amount of vehicles are practical for platoons managed by a sole authority like a logistic company. But these constraints prevent platooning from being used to its fullest potential. Ideally, platoons are created and managed dynamically whenever possible to maximize spacial and economic effects. In order to enable dynamic platoons, vehicles which have never met before would have to cooperate. This, on the other hand, challenges the implicit trust relationship, which was taken as granted in previous research projects. It cannot be ensured that all platooning-capable vehicles are functioning flawlessly or act without malicious intent. Since data sent by predecessors in the platoon needs to be processed by the cruise controller and thus directly affects the acceleration of the follower vehicle, safety is of concern [7].

To reduce the safety risks in dynamic platoons, a host vehicle needs to be able to determine, if members of its own platoon are acting as expected and if the data they send can be *trusted*. In this paper we propose the concept of Trust in Platoons (TriP). Therefore, we provide the necessary definition of trust and how it can be calculated by platoon members. This includes which criteria, like consistency of velocity, acceleration or reported distance to neighbors, are important for platooning and need to be considered. Additionally, a concept for verifying their accuracy is developed and those criteria are combined to create a single, momentaneous rating for each interaction. Finally, the ratings are aggregated into a score, which models the evolution of trust including aging and betrayal in order to have a stable but also responsive estimate for the cooperativeness of the other platoon members.

As large scale physical experiments with platoons are infeasible, the only possible method to implement and evaluate TriP and its effectiveness was to simulate it. Basic platooning concepts are already implemented by PLEXE [8], a platooning extension for the vehicular network simulator Veins [9]. This work was part of a project using Artery [10]. Hence for the evaluation, we ported PLEXE to Artery, which bases on Veins and additionally includes all upper layer protocols of ETSI ITS-G5 [11]. However, TriP is not limited to ITS-G5 and could be used with all V2X communication technologies.

The remainder of this paper is organized as follows: Section II gives an overview of previous work in this domain before Section III introduces the model used in TriP, which is evaluated in Section Section IV. Section V concludes the paper and gives a short outlook.

This research was partially supported by the BMBF project SecForCARs (funding number 16KIS0791) and the DFG Research Unit Controlling Concurrent Change (funding number FOR 1800).

[1] Authors are with the Technische Universität Braunschweig, Institute of Operating Systems and Computer Networks, 38106 Braunschweig, Germany {garlichs,willecke,wegner,wolf}@ibr.cs.tu-bs.de

II. Related Work

Amoozadeh et al. [12] presented sophisticated and detailed approach to manage platoons. In their approach the leader is the central management unit of the platoon, and followers only issue maneuver requests directly to it. Security is not taken into account so that neither authentication nor confidentiality can be provided, and thus attacks against the platoon are possible. Also, malicious vehicles cannot be excluded from the platoon, because trust and reputation are not considered.

Establishing trust while keeping privacy has been studied in other areas such as participatory sensing. What, e.g., Wang et al. [13] propose is not suitable for platooning because it models trust only unidirectional where the server calculates trust to its clients whereas the trust has to be mutual in platooning.

Timpner et al.'s work in [14] helps vehicles to build a trust rating for each other by forming so called Parking Communities that help to find free parking spaces. It relies on frequent re-encouters between the community members. We deem this assumption unrealistic in most situations as even on regularly driven routes, two vehicles rarely drive on the same road, at the same time, at a similar pace and in close proximity to each other. Hence, this not applicable for platooning.

The Byzantine Agreement Service of Xu et al. [15] could be used to asses a platoons current speed, but it is not directly applicable to detect actual misbehavior and react accordingly to the respective entities.

Hu et al. presented a trust management architecture for platooning in [16]. Trust is managed by a central trusted authority, where human participants can rate the quality of service by a human platoon leader, resulting in a reputation score for each leader, which can be queried before joining a platoon. Their approach does not consider malfunctions or malicious behavior and mainly focuses on quality of service for platooning. Due to their centralized approach, it does not scale well and heavily depends on infrastructure. Also privacy of users is not considered. In [17], they improved their approach by using weighted and filtered user feedback to increase stability and security of their leader's reputation rating. Still, scalability and privacy issues persist.

DeBruhl et al. [18] simulated attacks and abnormal behavior in vehicle platoons using the CACC. To detect their conceived attacks they designed a misbehavior detection modelling the controller of each platoon member locally to determine anomalies. Upon detection, their vehicles switch to Adaptive Cruise Control (ACC) to avoid collision. The model is able to detect misbehavior of members, but is susceptible to noise when the platoon legitimately accelerates.

A simulation study by van der Heijden et al. [19] demonstrated how a platoon member is able to cause harm to other members. They built a scenario for the popular platooning simulation environment PLEXE [8] where an attacker starts to inject bogus data in its beacons. These attacks were evaluated against multiple cooperative cruise controllers. The results show all controllers being susceptible to the injection of false data and thus the attacker is able to provoke collisions.

III. Modeling Trust

In order to prevent the collisions induced by malicious data that have been shown in [19], only accurate data can be passed to the cruise controller. Since a host vehicle cannot judge the value of a single data point received in a beacon, it has to be compared to the sender's actual behavior over a longer period of time. This way, the host vehicle can gain more and more *trust* in the sender's benignity or detect its misbehavior. Depending on the trust in the predecessor in a platoon, the host vehicle can then for example regulate the safety gap to that vehicle. If the trust falls below a certain threshold, it could ultimately decide not to be in a platoon with that vehicle at all. This section introduces the details of TriP - our trust-based misbehavior detection system for platoons.

Generally, trust towards an entity can be described as the aggregation of evaluations of multiple interactions with that entity. TriP has three steps: First, a variety of criteria are considered in order to evaluate a single interaction. Then, the outcomes of those evaluations are aggregated to one *trust score*, representing the accumulated history of interactions. Finally, there has to be a reaction towards other platoon members based on their respective trust scores.

A. Interactions

A host vehicle running automated driving functions, e.g. CACC, has to ultimately trust its own sensor data. When judging other platoon members' trustworthiness, it uses that data as ground truth to compare the others' actual behavior to the one they announced in their beacons. This comparison can hence only be done for vehicles in the perception range of the host vehicle's local sensors and of course works best for the direct predecessor and successor. It indicates the trustworthiness of those vehicles for different criteria. The evaluation of each criterion results in a value between 0 and 1 with 0 being the worst and 1 being the best possible score. The following trust criteria are used in TriP:

1) Velocity: The host vehicle is able to rate how well any member V_y's reported velocity matches the pace set by the leader V_L. Therefore, it calculates the reference velocity V_{ref} by considering V_L's last reported velocity v_L adjusted by the reported acceleration a_L and the time b_L which passed since the last leader beacon was received:

$$v_{ref} = v_L + b_L \cdot a_L \tag{1}$$

Then, it calculates the error of V_y's reported velocity v_y to V_{ref}. The error is subtracted from 1, reflecting the divergence from the perfect behavior. $v_{y,ref}$ denotes how well V_y's reported velocity matches the leader's velocity.

$$v_{y,ref} = \begin{cases} max\left(1 - \left|\frac{v_y - v_{ref}}{v_{ref}}\right|, 0\right), \text{ if } v_{ref} > 0 \\ max\left(1 - |v_y\ |, 0\right), \text{ else} \end{cases} \tag{2}$$

In case of determining the velocity criterion for the platoon leader itself, the host vehicle needs to consider another reference velocity, as the leader always matches its own velocity. Such a reference velocity could depend on the speed limit and traffic density, to represent how well the leader sets the pace for the platoon. This could be achieved by comparing the leader's velocity to an average velocity of other vehicles travelling in the same direction or performing an outlier detection [20]. This way, platoon members can compute trust samples for misbehaving platoon leaders, which are e.g. slowing down the progression of the platoon.

2) Distance: To rate how precise the preceding vehicle V_y can locate itself and how plausible the reported position is, the host vehicle first calculates the distance between its own location and the position reported in V_y's last beacon, resulting in $d_{b,y}$. Then it calculates the error to the distance d measured by its own sensors, e.g. radar, and subtracts the error from perfect result:

$$d_y = max\left(1 - \left|\frac{d_{b,y} - d}{d}\right|, 0\right) \tag{3}$$

3) Acceleration: To evaluate the received acceleration, the host vehicle first calculates the difference of the received acceleration to its own acceleration to get a_{diff}. The host vehicle determines the relative velocity v_{rel} by calculating the derivative of the change in distance d and the sample interval t_s. Finally, the derivative of the relative velocity v_{rel} is calculated and multiplied it with the difference in acceleration a_{diff}:

$$a_y = max\left(1 - \left|\frac{v_{rel}}{t_s} \cdot a_{diff}\right|, 0\right) \tag{4}$$

4) Jerkiness: The absolute jerk for vehicle V_y is calculated by deriving the indicated acceleration values between the last and current beacon. High jerkiness of the predecessor vehicle is not only a safety risk, it may also bear discomfort to the host vehicle's passengers, since it has to react to an abrupt change in acceleration as well. Because every maneuver induces at least some amount of jerk, it is only credited negatively once it exceeds the threshold j_{thres}. The more the absolute jerk $j_{y,abs}$ of V_y exceeds the threshold, the lower the result for the criterion:

$$j_y = min\left(\frac{j_{\text{thres}}}{j_{y,abs}}, 1\right) \tag{5}$$

5) Beacon Delivery Timeout: Each member of a platoon is expected to broadcast a beacon regularly. When a member vehicle V_y does not send the expected beacon until a timeout, it is not possible to evaluate the aforementioned criteria and the criterion to_y is set to 0, indicating non-cooperativeness. When receiving a beacon, to_y is set to 1.

B. Trust Samples

So after either exceeding that timeout or after receiving a beacon from V_y, all criteria are combined to one single *trust sample* t_y, rating the respective interaction, as follows:

$$t_y = to_y \cdot v_{y,ref}^{w_v} \cdot d_y^{w_d} \cdot a_y^{w_a} \cdot j_y^{w_j} \tag{6}$$

When the deadline was reached, the timeout factor to_y determines the whole sample as invalid. Each other criterion is weighted separately with the exponents $w_v, w_d, w_a, w_j \in \mathbb{R}$ respectively to determine its influence. When the weight equals 1, the respective criterion remains unchanged. If it is less than 1, the criterion impacts the trust sample less and it is more impactful, if the weight exceeds 1. The weighting is chosen as exponential to be able to compensate for smaller differences due to inaccuracy of measurements, but still to be able to impact the trust sample rigorously when the received data diverges. The weighted criteria are then multiplied so that even a single criterion is able to influence the resulting trust sample.

C. Building Trust

Since the resulting trust samples for interactions may change over time and a trust is build over repeated interactions, these values need to be aggregated in a way that allows historic evolution, judgment on reliability and certainty. Thus, the aggregation can be represented in a probability distribution function (PDF) by using the framework of *Bayesian statistics*.

Bayesian trust systems from literature are often based on the *Beta Reputation System* [21] which utilizes the beta PDF [14]. The Beta Reputation System is only able to express ratings in two outcomes, such as *trusted* and *untrusted*. The *Dirichlet Reputation System* [22] however enables more fine granular ratings compared to the Beta Reputation System, as is is based on the Dirichlet PDF which is the generalization of the Beta PDF, supporting an arbitrary number of outcomes.

To describe the trustworthiness, $k = 5$ mutually exclusive, equidistant and ascending trust rating levels $\in [0, 1]$ are defined as *untrustworthy*, *bad*, *acceptable*, *good* and *excellent*.

When vehicle V_x calculates a $t_y \in [0, 1]$ for V_y, it is mapped to a k-dimensional vector r_y^x where every element of the vector is 0 except the one for the respective trust rating level (which is set to 1). The vector r_y^x can be interpreted as probability distribution with a single outcome, determined by the trust sample.

Multiple outcome vectors r_y^x from V_x for V_y are then aggregated in the rating vector:

$$R_y = (R_y(1), \ldots, R_y(k)) \tag{7}$$

It accumulates all past outcomes as *a postiori* distribution, where $R_y(i)$ denotes the accumulation of outcomes for the i-th trust rating level. The aggregation for ratings is defined recursively using the scalar aging factor $\lambda_y \in [0, 1]$ as:

$$R_{y,(t+1)} = \lambda_y \cdot R_{y,t} + r_y^x \tag{8}$$

λ_y defines how much previous samples are taken into consideration. With $\lambda_y = 1$ old ratings live forever and with $\lambda_y = 0$ old ratings are forgotten with each new rating. To increase the impact of low trust, λ_y needs to depend on the final *trust score* for V_y denoted as T_y, defined by Equation (12):

$$\lambda_y = 1 - (T_y \cdot w_t) \tag{9}$$

$w_t \in [0,1]$ denotes a linear weighting constant, influencing the overall impact of T_y within λ_y. Vehicles can lose trust more quickly, when sending incorrect data. Also, vehicles with low trust need longer time to recover trust. This behavior is chosen to model social trust relations, where single action can deal significant damage to a trust relationship, which it is hard to recover from.

The probability distribution vector S_y represents the current trust rating vector for V_y:

$$S_y = (S_y(1), \ldots, S_y(k)) \tag{10}$$

Therefore, the *expected value* for each element of the aggregated rating vector R_y and an *a priori* distribution vector a weighted with C is calculated according to [22] as:

$$S_y(i) = \frac{R_y(i) + C \cdot a(i)}{C + \sum_{j=1}^{k} R_y(j)} \tag{11}$$

When no *a priori* distribution vector a is known, each outcome is assumed equally likely, resulting in $a(i) = 0.2$ for each of the $k = 5$ trust level. Since the vector S_y can also be interpreted as a probability distribution vector, it could be exchanged with other vehicles an serve as a prior a itself.

Lastly, to calculate a single *trust score* $T_y^x \in [0,1]$ for V_y by V_x, the expected value for each trust level are multiplied with the their significance and summed up as:

$$T_y^x = \sum_{i=1}^{k} \frac{i-1}{k-1} \cdot S(i)_y \tag{12}$$

D. Misbehavior Detection

The trust score resulting from the sample and aggregation models indicates how accurate, well-behaved and reliable a member of a platoon is perceived by the host vehicle. This way misbehavior can be identified and defense mechanisms against attacks can be deployed.

Therefore, TriP can improve the passengers' safety by adjusting the distance to the predecessor inversely proportional to the trust score to compensate for their deficiencies. Ultimately, low trust could lead to changes in the platoon itself as members could decide to leave when being part of the platoon is deemed as too risky due to imprecise sensor data or an uncomfortable driving experience.

When a trust score is known beforehand it can also impact platooning protocol decisions. An ego-vehicle might not want to follow an untrustworthy leader or a leader might not affiliate an untrustworthy vehicle into its platoon.

Therefore, the host vehicle periodically evaluates the trust score of its direct predecessor. When the score is *excellent* (above 0.8), the configured platooning spacing d_{space}, set by the leader, is used. When the score ranges between *good* (≤ 0.8) and *bad* (≥ 0.2), the safe spacing D_{safe} to its predecessor is calculated using the current velocity and constant time gap T_{acc}, used by the ACC:

$$D_{safe} = v \cdot T_{acc} \tag{13}$$

Then, the final distance to the predecessor is calculated as follows:

$$D(T_y) = d_{space} + ((D_{safe} - d_{space}) \cdot (0.8 - T_y)) \tag{14}$$

Fig. 1: Platoon formation in the simulation

TABLE I: Trust Model Configuration

Symbol	Description	Value
w_v	Velocity weight	4.0
w_d	Distance weight	1.0
w_a	Acceleration weight	2.0
w_j	Jerk weight	1.0
w_t	Trust weighting constant	0.85
C	A priori distribution weight	0.2
T_{acc}	ACC time gap	$1.2s$

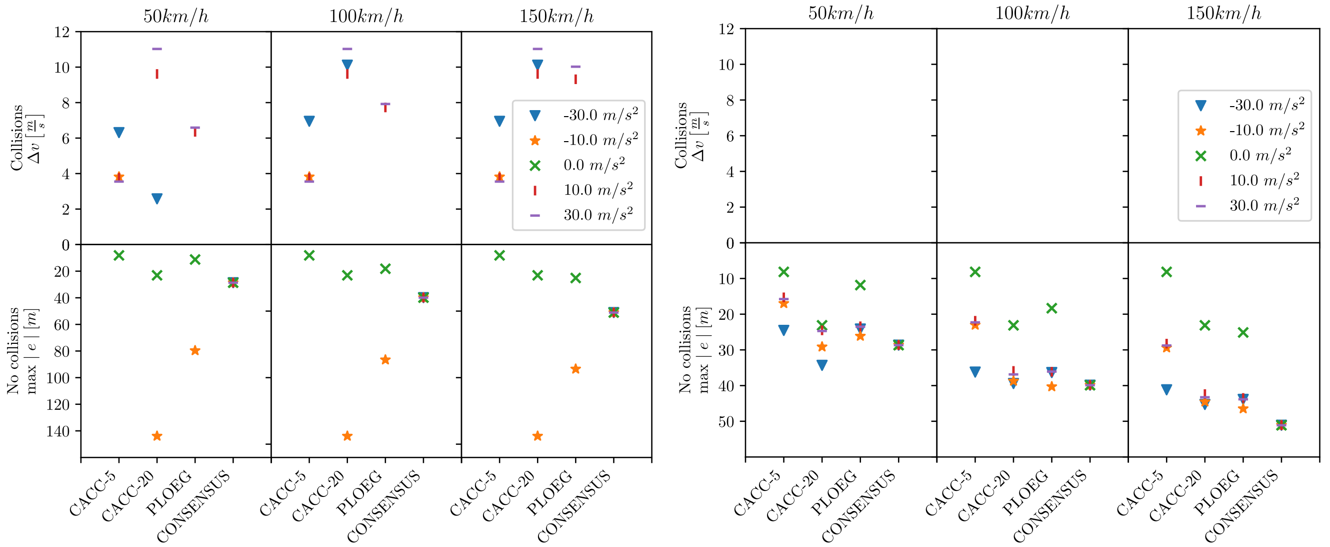
Hence, the trust score adjusted distance ranges from the CACC distance to the ACC distance. In this stage, it is still possible for the host vehicle to regain trust in its predecessor. Thus, when trust is *excellent* again, its spacing is reset to default.

When the trust score dips below *bad* to *untrustworthy* (e.g. ≤ 0.2), the ego vehicle switches from the CACC to the ACC car-following model. It slowly increases the headway distance linearly, effectively detaching itself from the platoon, removing the dependency on its untrustworthy predecessor. In this case an extra, out-of-order beacon will be sent to inform its own successors of its own upcoming drastic change in behavior.

IV. EVALUATION

To determine TriPs effectiveness, it was implemented for PLEXE and evaluated against attacks from literature and different platoon controllers. Van der Heijden et al. showed the impact of various attacks of platoon insiders on different platoon controllers using PLEXE [19]. We ported PLEXE to Artery to facilitate platooning using ITS-G5 communication as well as the attacks proposed by van der Heijden et al.. The validity of the port was verified by reproducing previously published results (compare Figure 2a to Figure 5 of [19]).

The attacks were performed in a scenario were a platoon of 8 vehicles with a set speed is spawned on a highway as shown in Figure 1. Five seconds into the simulation, the platoon leader starts to oscillate its velocity with $0.2\,Hz$ and an amplitude of $10\,\frac{km}{h}$. As the platoon is spawned in a stable state, followers adapt to the oscillation, maintaining the distance.One simulation run lasts for $60\,s$. From $t = 30s$ onwards, the attacker (V_3) injects either wrong acceleration, velocity or position data in its beacons. Each attack was conducted with four platooning controllers (CACC [2] with 5 and 20m safte gap, PLOEG [23] and CONSENSUS[24]), provided by PLEXE, at three distinct set speeds with different spacing. Then, each injection attack was evaluated using false data of different magnitudes. The attacks are evaluated by impact on platoon stability (error of inter-vehicle distance) and, in case of crashes, their effectiveness to cause harm (velocity difference). Figure 2a shows how the injection of false data is able to cause harm. A collision is provoked in most cases indicated by a mark in the upper half of the plot, showing its impact.



(a) Acceleration attack proposed by [19] without TriP

(b) Acceleration attack with TriP and defense mechanism

Fig. 2: Acceleration Injection Attack: Attacker starts to maliciously inject acceleration data in its platooning beacons

We implemented the counter measures discussed in Section III-D and configured TriP according to Table I. Since the manipulation of velocity and acceleration where originally the most devastating attacks, both trust criteria are weighted heavier than distance and jerk criteria. In the beginning, all trust scores are assumed equally likely. The trust weighting constant w_t is chosen as 0.85 so that even highly trusted vehicles can be degraded quite fast.

Then, all attack scenarios were repeated with the same configurations to evaluate how trust scores develop, when V_3 attacks the platoon in different ways, and how the victims react to the attack. While Figure 2a demonstrates the success of the attacks if no countermeasures were applied, Figure 2b illustrates the ability of TriP to reduce safety issues.

Most notably, all crashes the attacker was able to cause without the use of TriP are detected and prevented by the attacker's successor as none of the marks are in the upper half and only distance errors remain. When the victim is attacked, its implementation of the trust model identifies the mismatch between the attacker's beacons and its actual behavior. Hence, the respective trust criterion is reduced, creating low trust samples for mismatching beacons. Therefore, the overall trust score for the attacker drops and the victim starts to deploy its countermeasures. These increase the safety distance towards the attacker and eventually eliminate the reliance on the attacker's data, when the victim switches to ACC. Crashes which occurred previously are converted to an error in distance, because the spacing between the attacker and the victim exceeds the inter-vehicle spacing set by the leader. Ultimately, the attacker's efforts to cause accidents with the propagation of false data are completely nullified by TriP. Similar to the acceleration injection attack, all collisions caused by speed and position injection attacks were successfully prevented and converted to a distance error as well. However, the plots looked very similar and were omitted due to space limitations.

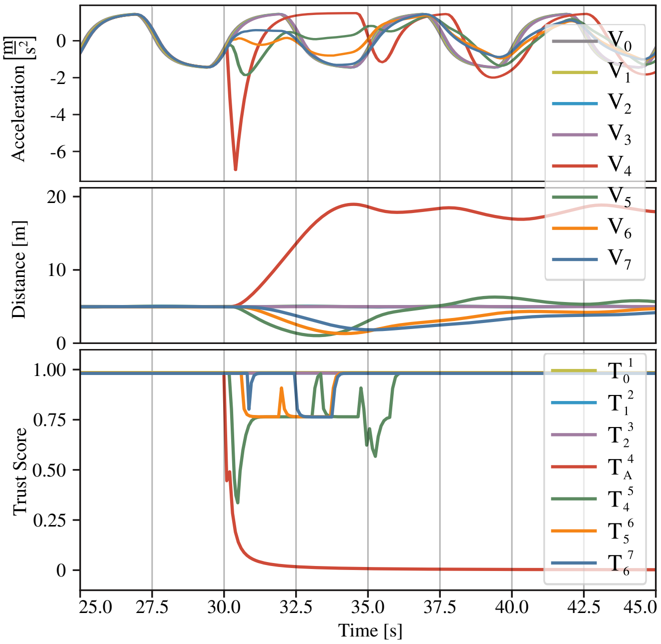


Fig. 3: Simulation results for acceleration injection attack carried out by V_3 with $-30\,\frac{m}{s^2}$ at $t = 30\,s$. Victim V_4 detects the attack by V_3, as shown by T_A^4, and reacts by increasing the distance to its predecessor.

One distinct case is shown in Figure 3, where the attacker tries to cause an accident by sending a highly negative acceleration in its beacons indicating an emergency brake at a platoon velocity of 150 $\frac{km}{h}$ and an inter-vehicle spacing of $\;5\,m$. It can be observed that the first victim immediately starts to brake as hard as Artery's traffic simulator allows ($-7.5\,\frac{m}{s^2}$), as it assumes an emergency brake by the attacker. A vehicle with good brakes and under good environmental conditions can reach a deceleration of 10 $\frac{m}{s^2}$ when performing an emergency stop [25]. With the next beacon, the victim detects a mismatch of the attacker's announced and its own measured deceleration and the increase of relative acceleration. Due to the high jerk and the acceleration mismatch the victim reduces its score for the attacker to untrustworthy quite fast and thus increases the safety distance towards the attacker. Figure 3 also shows the increases of distance, too, based on reduced trust. Since leader beacons still impact the control loop, the distance is not linear, but also follows the sinusoidal velocity targeted by the leader. The increase of distance causes platoon instability, as it pushes back the victim's followers. Since the mismatch of acceleration persists, the attacker's score stays low and the victim increases the safety distance over time.

Due to the high jerk, vehicle 5, 6 and 7 also calculate low trust scores for their respective predecessors, reducing their rating to good and acceptable temporarily. The acceptable rating for vehicle 4 shows that initially it is quite difficult for vehicle 5 to differentiate between an attack and the defense mechanisms of the victim.

When the trust scores stabilize again after the initial attack, all vehicles except for V_4 converge to the initial inter-vehicle spacing of $5\,m$. The platoon is split in two parts (V_0 - V_3 and V_4 - V_7). As the non-congruent acceleration shows, the second half is rather unstable, but still safe as no crash occurred. In this stage it would be feasible for the second half of the platoon to undertake further actions. For example, V_4 leaving the old platoon and declaring itself the new leader and thus, stabilize the platoon again.

V. Conclusion

When vehicles drive in tightly coupled formations like a platoon, they take actions based on data provided by others. Those actions can include accelerating, steering and braking and hence are highly safety critical. Therefore, it is crucial that a vehicle can *trust* the data it receives from others. In this paper we proposed TriP - a detailed model for the trust relationship in such platoons. It examines the behavior of other platoon members and derives a trust rating based on the Dirichlet Reputation System. In order to evaluate the effectiveness of that model, the trust model was implemented in the simulation framework Artery. All the attacks to the platoons that lead to collisions in [19] could successfully be detected early enough to prevent the collisions. This proves the great benefit such a system can provide in terms of functional safety. Because the information gathered about misbehaving vehicles is highly relevant for other potential platooning partners of those vehicles, we are planning to gather the locally generated trust ratings in a central reputation system. The key challenge of that system is to accurately inform about misbehavior while preserving everyone's privacy.

References

[1] B. Van Arem, C. J. Van Driel, and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.

[2] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[3] A. Vinel, L. Lan, and N. Lyamin, "Vehicle-to-Vehicle Communication in C-ACC/Platooning Scenarios," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 192–197, 2015.

[4] L. Li and F.-Y. Wang, *Advanced Motion Control and Sensing for Intelligent Vehicles*. Springer Science & Business Media, 2007.

[5] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated Vehicle Control Developments in the PATH Program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[6] P. Jootel, "SAfe Road TRains for the Environment," *SARTRE Project, Final Project Report*, 2012.

[7] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular Platooning in an Adversarial Environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, 2015, pp. 167–178.

[8] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "PLEXE: A Platooning Extension for Veins," in *IEEE Vehicular Networking Conference (VNC)*, 2014, pp. 53–60.

[9] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[10] R. Riebl, H.-J. Günther, C. Facchi, and L. Wolf, "Artery - Extending Veins for VANET Applications," in *4th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS 2015)*. Budapest, Hungary: IEEE, Jun 2015.

[11] A. Festag, "Cooperative Intelligent Transport Systems Standards in Europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[12] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon Management with Cooperative Adaptive Cruise Control Enabled by VANET," *Elsevier Vehicular Communications*, vol. 2, no. 2, pp. 110–123, 2015.

[13] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTSense: Anonymous Reputation and Trust in Participatory Sensing," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2013, pp. 2517–2525.

[14] J. Timpner, D. Schürmann, and L. Wolf, "Trustworthy Parking Communities: Helping your Neighbor to Find a Space," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, pp. 120–132, 2016.

[15] W. Xu, M. Wegner, L. Wolf, and R. Kapitza, "Byzantine Agreement Service for Cooperative Wireless Embedded Systems," in *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2017, pp. 10–15.

[16] H. Hu, R. Lu, and Z. Zhang, "TPSQ: Trust-Based Platoon Service Query via Vehicular Communications," *Springer Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 262–277, 2017.

[17] H. Hu, R. Lu, Z. Zhang, and J. Shao, "REPLACE: A Reliable Trust-Based Platoon Service Recommendation Scheme in VANET," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1786–1797, 2017.

[18] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is Your Commute Driving You Crazy?: A Study of Misbehavior in Vehicular Platoons," in *Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 22:1–22:11.

[19] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing Attacks on Cooperative Adaptive Cruise Control (CACC)," in *IEEE Vehicular Networking Conference (VNC)*, 2017, pp. 45–52.

[20] M. Matousek, M. Yassin, R. van der Heijden, F. Kargl *et al.*, "Robust Detection of Anomalous Driving Behavior," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[21] A. Josang and R. Ismail, "The Beta Reputation System," in *Proceedings of the Bled eCommerce Conference*, vol. 5, 2002, pp. 2502–2511.

[22] A. Josang and J. Haller, "Dirichlet Reputation Systems," in *IEEE International Conference on Availability, Reliability and Security (ARES)*, 2007, pp. 112–119.

[23] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of CACC Performance Subject to Unreliable Wireless Communication," in *IEEE International Conference on Intelligent Transportation Systems*, 2013, pp. 1210–1216.

[24] M. Di Bernardo, A. Salvi, and S. Santini, "Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time-Varying Heterogeneous Communication Delays," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 102–112, 2015.

[25] H. Burg and A. Moser, *Handbuch Verkehrsunfallrekonstruktion: Unfallaufnahme, Fahrdynamik, Simulation*. Springer-Verlag, 2009.