Model-based Membership Verification in Vehicular Platoons

Mikael Asplund
Department of Computer and Information Science,
Linköping University, SE-581 83 Linköping, Sweden
mikael.asplund@liu.se

***Abstract*—Cooperative vehicular systems have the potential to significantly increase traffic efficiency and safety. However, they also raise the question of to what extent information that is received from other vehicles can be trusted. In this paper we present a novel approach for increasing the trustworthiness of cooperative driving through a model-based approach for verifying membership views in vehicular platoons. We define a formal model for platoon membership, cooperative awareness claims, and membership verification mechanisms. With the help of a satisfiability solver, we are able to quantitatively analyse the impact of different system parameters on the verifiability of received information. Our results demonstrate the importance of cross validating received messages, as well as the surprising difficulty in establishing correct membership views despite powerful verification mechanisms.**

I. Introduction

Platoons of vehicles represent an instance of cooperative intelligent vehicles that have reached a relatively high maturity level [4]. Already today, vehicles are able to use radar technology to maintain a constant distance to the vehicle in front and thereby reduce fuel consumption. Moreover, an inter-vehicle communication standard for platooning is currently being developed by ETSI [10].

We argue that secure group membership is a crucial component for future vehicular coordination systems. Accurate and up-to-date membership views are needed in platoons for two main reasons, (1) reliable group communication (2) for the leader to make safe and appropriate driving decisions. The key challenges to implementing accurate membership for vehicular environments are the unreliable nature of wireless communication, the high level of dynamism in vehicle movements, interaction with non-automated vehicles [22], and the increased security risks associated with external communication interfaces.

Current vehicular systems have been shown to be vulnerable to unauthorised remote access [6]. Automotive manufacturers compete with features ranging from advanced driver assistance to social networking and third party applications (e.g., Ford Sync), thereby further increasing the attack surfaces. Potential incentives for attacks can range from financial gain (blackmailing) to political motives and vandalism. In addition to security measures that prevent core systems from being compromised, the philosophy of defence in depth tells us that we should also design cooperative algorithms to be robust in presence of compromised (or *faulty*) vehicles. Therefore, we consider an arbitrary (or Byzantine) node fault model which includes both unintentional and intentional (malicious) faults. Note that a vehicle can be faulty due to harmful code without the driver's knowledge.

The problem of platoon membership views is very much related to the concept of membership in distributed systems [7], but there are also differences. Consider the scenario depicted in Fig. 1. There are three vehicles A,B,D travelling in a formation. However, the last vehicle D has received its membership view from another faulty vehicle C. If we were to ignore the vehicular domain and consider this system as just four nodes in a distributed system, this could be seen as two separate groups, AB and CD (since D joined the group announced by C). However, in the vehicular domain this situation is potentially dangerous since the braking action of A might not be properly propagated to vehicle D. This shows that it is desirable for the platoon membership view to correspond to the physical formation of vehicles.

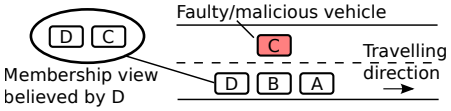

Fig. 1. Scenario with an incorrect platoon membership view

This scenario can be prevented in several ways. One option is that platoon groups are formed manually by drivers that have reason to trust each other (e.g., if employed by the same company). Another option is to allow dynamic formation of platoon groups, but require a human to supervise and verify all membership changes. A third option would be to prevent the possibility of any vehicles ever acting maliciously (e.g., by requiring certified software and trusted platform modules [12] to prevent execution of unauthorised code). However, this does not prevent cases where the incorrect membership view is a result of a non-malicious fault. It also requires that encryption keys are properly managed and are not compromised as seems to have happened in other domains [20]. Finally, specific mechanisms can be used to detect false position information [15] under some circumstances.

In this paper we ask the question: under which conditions is it possible for a vehicle in a platoon to automatically verify that the membership view that it has received from the platoon leader corresponds to the physical formation of vehicles?

The purpose of this investigation is to assess the conceptual mechanisms that can be used to verify the correctness of membership views. Moreover, we believe that the ability to reason about membership views in a stringent manner could also be useful as component in a security framework, similar to what has been done in other fields [18].

Our approach is based on the idea of performing a case-based analysis of the information that is received by a vehicle. Consider the situation depicted in Fig. 2. We perform the reasoning from the point of view of one particular vehicle called the *ego vehicle* (i.e., the vehicle for which the assessment is performed). The use of an ego vehicle is a common method for modelling behaviour of vehicles interactions (e.g., see [1]). The ego vehicle is not to be considered a special vehicle, but as a representative of any vehicle capable of participating in a vehicular platoon. Vehicles receive information from surrounding vehicles through *cooperative awareness claims*, and platoon members receive a platoon membership view from the platoon leader. The vehicles will also have some on-board sensors like distance radar and cameras. In Section II we describe an abstract and formal model of such a system. The information received by the ego vehicle should normally be consistent, but there might be cases where erroneous data is received. To detect such cases, it is reasonable to have verification mechanisms that can be used to ensure that the received information is indeed correct. In Section III we describe six verification mechanisms for this purpose.

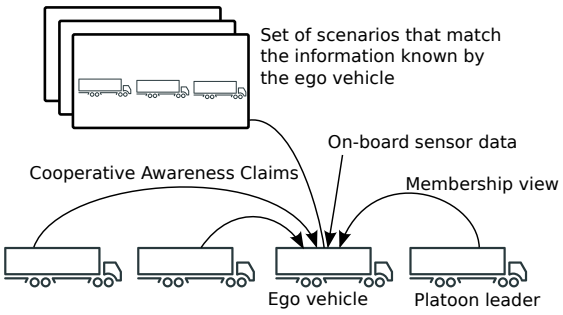

Fig. 2. Conceptual view of the information received by the ego vehicle

However, even if the verification tests are passed, the membership view might still be incorrect if some of the vehicles in the vicinity are faulty, thereby causing them to convey false information. Thus, if the ego vehicle is suspicious (as it should be), it could potentially consider a number of cases that all match the received information but where the membership view is incorrect (depicted in the figure as the set of matching scenarios). In Section IV we present a modelling framework that provides this type of reasoning. In particular, for a given set of inputs, this framework can enumerate all the scenarios that match the input. The more information that can be gathered and verified about the environment, the fewer matching scenarios will be found. In the ideal case, there is just one possible scenario (i.e., the membership view is correct by necessity). In Section V we show how the various verification mechanisms impact the number of matching solutions.

To summarise, the contributions in this paper are threefold.

- A formalisation of platoon membership views and verification mechanisms with which a vehicle can validate these views.
- A basic constraint-based framework for enumerating all physical solutions that match the information available to a vehicle.
- An experimental evaluation of how factors such as length of membership views, and different verification mechanisms affect how confident a vehicle can be that its view of the world is indeed correct.

II. System model

In this section we present the system model and explain our basic assumptions. We have consciously avoided making the model very complex in terms of time and space. Our focus is on the basic properties of membership views and how a vehicle can verify that the view is correct. On a high level we model the system as a tuple (V, M, A, P, C) as follows.

- V - an unbounded set of vehicles
- M - an unbounded set of membership views
- A - an unbounded set of cooperative awareness claims
- F - a set of functions
- C - a set of constraints

The sets V, M, A represent the subjects in the model. The model also includes constants[1] of these types with special meaning, like the ego vehicle $e \in V$. The functions in F allows Boolean predicates and composite properties to be defined for the object domains, for example the model contains a Boolean predicate $\texttt{correct}(v)$ that denotes whether a vehicle v is correct or faulty. Finally, the set C contains the constraints which can be used to include factual statements and assumptions in the model. For example, our model include the constraint $\texttt{correct}(e)$, where e is the ego vehicle, which means that the ego vehicle is assumed to be non-faulty. The constraints are expressed as formulae in first-order predicate logic.

We now proceed to give a more detailed explanation of the model, by considering vehicles, membership views and cooperative awareness claims separately.

A. Vehicles

Constants In this paper we are interested in a fairly specific scenario, namely a vehicle which is part of a platoon that wishes to verify that the membership view it has received is correct. In addition to the ego vehicle e introduced above, the model includes an array of vehicles $f = [v_1, \ldots, v_n]$, that represent the physical formation which the ego vehicle belongs to. The vehicles that are not part of the formation f can also be formed in platoons, but we do not model this explicitly. Finally, there is a constant $\texttt{nil} \in V$ which is used when representing arrays with variable length.

Functions There are essentially four functions related to the vehicle domain V.

[1] formally a function with arity 0

- `absolutePos`: $V \to \mathbb{N}$, denotes the longitudinal position of the vehicle. We have chosen a discrete value domain, as it suffices for our high-level analysis of the problem.
- `vehicleInFront`: $V \to V$, if a vehicle is part of the formation, this function denotes the vehicle in front of it. For the leader vehicle the value is of this function is nil.
- `vehicleBehind`: $V \to V$, if a vehicle is part of the formation, this function denotes the vehicle behind it. For the last vehicle the value of this function is nil.
- `correct`: $V \to \{0, 1\}$, denotes whether a vehicle is correct or faulty.

Constraints The model includes a number of physical constraints that ensure that the vehicles in the formation have physical positions that are consistent with respect to each other, that there is no overlap between vehicle positions, and no other vehicles drive too close to the formation. Other than this, there are no constraints on the positions of vehicles not in the formation. We conceptually consider them all to be in another lane.

As previously mentioned the ego vehicle is assumed to be correct. Moreover, the ego vehicle is always part of the formation f.

B. Membership views

The membership views provide the participating vehicles with information of the platoon configuration. In our current model, these views always originate from the platoon leader. Moreover, we do not consider the process of actually forming membership views, see [13], [22] for more on this topic.

Functions There are five functions related to the membership views.

- `member`: $M \times \mathbb{N} \to V$, denotes the actual members in the view. We model this as a binary function, that results in nil for all positions that are outside the range $[1, n]$ where n is an integer number that represents the *length of the membership view*.
- `sender`: $M \to V$, denotes the sender of the view. In the current model $sender(m) = member(m, 1)$, meaning that the sender is always the platoon leader.
- `believedView`: $V \to M$, denotes the membership view that a particular vehicle believes to be true.
- `consistentView`: $M \to \{0, 1\}$, this function checks whether a membership view is well-formed. For example, all vehicles should be distinct, and the view should contain at least one vehicle.
- `honestView`: $M \to \{0, 1\}$, this function asserts that the view is consistent with all the information known by the sender of the view.

Constraints There are two basic constraints related to membership views in our model:

- All `correct` vehicles in the formation f has a `believedView` that satisfies `consistentView` and where it is a `member`.
- If the `sender` of a membership view is `correct`, then the membership view satisfies `honestView`.

C. Cooperative awareness claims

The cooperative awareness claims represent an abstraction of the cooperative awareness messages that are part of the ETSI ITS standard [10] for inter-vehicle communication. Similarly to the membership views, we do not explicitly model time. A claim can be seen as the combination of all messages that have been received from a particular vehicle and that can still be considered as temporally valid.

Constants The model contains a single awareness claim constant `nilClaim` $\in A$, which represent lost messages (i.e., there are no recent enough messages that can be used as a claim).

Functions There are five functions related to awareness claims.

- `sender`: $A \to V$, denotes the sender of the awareness claim.
- `myAbsolutePos`: $A \to \mathbb{N}$, the claimed physical position of the sender.
- `myPlatoonPos`: $A \to \mathbb{N}$, the logical position in the platoon if the sender claims to be part of one.
- `myPlatoonLeader`: $A \to V$, the leader of the platoon that the sender claims to be part of.
- `receivedClaim`: $V \times V \to A$, denotes the awareness claim received by one vehicle from another. Note that this can be the `nilClaim` to allow for message loss.

Constraints The constraints related to cooperate awareness claims state that the information associated with a claim coming from a correct vehicle will also be correct.

D. Additional assumptions

In additions to the assumptions given as constraints in the basic model described above, we have added some constraints that can be seen as basic verification mechanisms.

- A correct vehicle knows if it is the first or last vehicle in a formation.
- All claims are signed so a vehicle cannot pretend to be another vehicle. This is consistent with the ETSI ITS standard that includes signed messages.
- A vehicle cannot send two conflicting claims.

We are well aware of the fact these are strong assumptions (in particular the third assumption) that themselves require appropriate security mechanisms. Relaxing them means opening up for even more attacks than what is already possible in our current model. Our objective is not to propose some particular mechanism that will solve all security problems related to platoon membership. Instead, we are assessing the relative strength of various verification mechanisms and demonstrate that even with strong assumptions on secure communication, it is difficult to completely rule out the possibility of incorrect membership views. Therefore, we believe that these assumptions are warranted.

Note that while these assumptions prevent sybil attacks [24], the fact that model allows for unbounded number of vehicles, one can easily mimic a sybil-attack in this model by adding more vehicles.

III. Verification mechanisms

In the ideal case, each vehicle has enough information from on-board sensors and surrounding vehicles to be able to verify and validate that the membership view it has received from the platoon leader is trustworthy. However, if the vehicle is not able to verify the trustworthiness of surrounding vehicles, and lacks high-quality information of its surroundings, then it might not be able to conclude that the membership view is really correct.

In this section we discuss six different mechanisms which a vehicle can employ to verify that a membership view is correct and trustworthy. Each of the mechanisms have different hardware requirements and might or might not be straightforward to implement in a real system.

A. Claim consistency and completeness

a) Verification of claim consistency: In our model a vehicle receives claims from surrounding vehicles and a membership view from the platoon leader. This claim consistency verification requirement dictates that for all the claims received by the ego vehicle, they match the membership view. In particular, the claimed position (both absolute position and platoon position) and the claimed platoon leader must match the membership view.

b) Verification of completeness: Given a membership view, the property of verified completeness holds if for every member of the view there is an awareness claim c that have been received by the ego vehicle. Note that this verification mechanism only ensures that the ego vehicle has received a claim from every vehicle in the membership view, not that their content is consistent.

Having this mechanism does not necessarily come without cost. With the current communication technologies and restrictions on antenna placement on vehicles, direct communication between all platoon members might not be possible with reasonable quality. To ensure full functionality, it has been proposed that messages from the platoon leader is propagated in a multi-hop fashion. In such a scenario, it would be much more expensive to ensure that all members receive continuous updates from all other members.

B. Verification of vehicle identity

Many instances of erroneous membership views can be prevented if the participating vehicles are able to check the identity of the vehicles around them. This could for example be achieved with the help of cameras and automatic license plate recognition. While there are many ways this particular technology could be compromised (e.g., false license plates) our focus is not primarily on the underlying technology but rather the effect of being able to identify vehicles.

We consider the ability to identify the vehicle in front and the vehicle behind as two separate mechanisms. This can for example be motivated by the fact that the trailer of a heavy-duty vehicle is a separate entity which is potentially owned by a different company than the truck owner.

C. Position-based verification

The third an final category of verification mechanisms that we have investigated relates to the physical position of vehicles. As vehicles are moving along the road, we are primarily interested in their relative position to each other.

c) Verification of distance to correct vehicles: Given a membership view m this mechanism verifies that for every pair of correct vehicles v_i, v_j in the view, the actual physical distance between v_i and v_j is the same as the distance between the claimed positions of the two vehicles.

d) Verification of inter-platoon distance: One of the attack types against platoon membership is that given a physical formation $f = [v_1, \ldots, v_n]$, multiple membership views are sent out, e.g., containing $[v_1, \ldots v_i]$ and $[v_{i+1}, \ldots v_n]$. Assuming that vehicles are able to receive membership views from nearby platoons, they can verify that for any two membership views, m_1 and m_2, the vehicles in those views are separated by a minimum distance. This also means that the vehicle would need to be able to overhear membership views of nearby platoons.

IV. Modelling framework

In this section we briefly describe the architecture and design of our modelling framework.

A. System overview

Fig. 3 shows the structural overview of our software framework, which is implemented using the Python language. The core of the framework is an assessment module. The input to this module is the system model constraints described in Section II, the verification mechanisms described in Section III, and finally what we chose to call a knowledge base. The knowledge base captures all the specific auxiliary information that is known about the system. For example, if we are interested in a scenario where the ego vehicle has received a membership view with two members, then we encode this as a constraint and add it to the knowledge base.

Given these inputs, the assessment module uses the Microsoft Z3 constraint solver to systematically reason about the existence of solutions that match the given constraints. The assessment module can be used both to count the number of matching solutions, as well as to generate a specific scenario and visualise it using the Matplotlib library.

B. Potential use cases

The purpose of our proposed modelling framework is twofold. First, it is useful as a research and design exploration tool to investigate the impact of various system design choices with regards to system trustworthiness. In this paper we use it primarily to evaluate the impact of various membership verification mechanisms.

We also envisage how this tool can be used as an integrated component in a platoon membership management software. In such a case, the "Knowledge base" described in Fig. 3 is dynamically instantiated with the actual information contained in messages received by a vehicle. Querying the model for

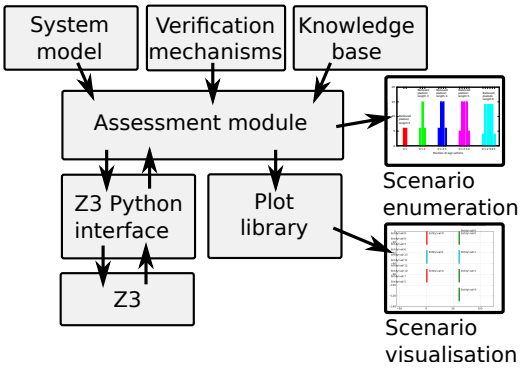

Fig. 3. Modelling framework overview

possible solutions that match the available information, there would be three possible outcomes:

- There are exactly 0 possible solutions. This means that the vehicle has received conflicting information, meaning that a fault or attack has occurred.
- There is exactly 1 possible solution. The vehicle has reason to trust the membership view. Provided that the modelling assumptions hold, there is no way that an attacker could have corrupted the membership view.
- There are more than 1 possible solutions. There is no obvious conflict that have been observed, but there is a possibility that the membership view is incorrect. Further action to rule out faults or attacks is recommended.

V. Evaluation

In this section we use the modelling framework to quantitatively assess the impact of varying key system parameters on the number of possible (faulty) scenarios. We begin the section by describing the experimental setup and the quantitative metrics we have used. We then consider three parameter types, the length of the membership views (as defined in Section II-B), the verification mechanisms and finally the number of faulty vehicles in the system.

A. Experimental setup

We configured the system to count the number of solutions that matched a given set of inputs. As detailed in the following subsection, we counted the number of solutions for different physical formation lengths and positions of the ego vehicle. In theory a formation can be of unbounded length, but in practice platoons cannot be arbitrarily long. Thus, we put a limit of the length of the physical formation to 7.

Unless otherwise stated, we performed these experiments with the assumption that the ego vehicle has received a membership view of length 3 and that the ego vehicle itself is in second position. Moreover, the default setting in the experiments is that the number of faulty vehicles is limited to 2. Table I provides a summary of these default parameters.

TABLE I
Default Evaluation Parameters

Parameter	Value
Max formation length	7
Membership view length	3
Ego position in membership view	2
Max number of faulty vehicles	2

B. Evaluation metrics

We now proceed to explain the metrics we used in the evaluation since they are in themselves quite complex.

Number of solutions For a given set of constraints, the constraint solver will either find a solution or not. By varying some of the input constraints that represent the physical reality in a systematic manner we can count how many representative scenarios there exist for a given configuration. In particular, we consider as input different lengths of the physical formation and the location of the ego vehicle within the formation. If we let $n = |f|$ be the length of the physical formation and $p \in [1, n]$ be real position of the ego vehicle within the formation, then a scenario is represented by the pair (n, p). We count the total number of solutions when varying (n, p). For a given maximum formation length $N = 7$ as we used in this evaluation, there can be at most $N(N+1)/2 = 28$ solutions.

While there might be many possible model instances (in some cases infinite) for a given pair (n, p), we only count 1 or 0, depending on whether there is at least one solution or not.

Number of microscopic solutions As a variation to the metric above, scenarios can be further differentiated. In addition to the parameters n and p, we add a mapping $c : \{1, \ldots, n\} \to \{0, 1\}$, that for each vehicle in the formation denotes whether that vehicle is correct (non-faulty). A scenario is thus represented by the triple (n, p, c). We define the number of *microscopic solutions* by varying this triple and counting how many of them that have a solution. Given that the ego vehicle is always correct, the total number of microscopic solutions for $N = 7$ is $\sum_{i=1}^{N} i2^{i-1} = 769$.

Shapley value Finally, we introduce a metric specifically to assess the various verification mechanisms. The problem is that the different mechanisms interact with each other in complex ways, meaning that one cannot easily evaluate them separately. To overcome this pedagogical challenge, we make use of a concept from game theory where the contribution of different players in a game can be quantified in a "fair" manner. In our case the players correspond to the verification mechanisms described in Section III. The Shapley value ϕ_i for a verification mechanism (or player) i can be defined as:

$$\phi_i = \sum_{S \subseteq \mathcal{V} \setminus \{i\}} \frac{|S|!(|\mathcal{V}| - |S| - 1)!}{|\mathcal{V}|!} (v(S \cup \{i\}) - v(S)) \quad (1)$$

where $\mathcal{V}$ is the set of all six verification mechanisms, $v(S)$ denotes the number of solutions that can be ruled out with the help of the subset of verification mechanisms S. Intuitively, the Shapley value calculates the average marginal contribution of each player for all possible orders of adding players. This formulation can be shown to have a number of desirable

properties for fairly dividing some particular cost. However, for the purpose of this paper, we are interested in this concept simply as a metric to assess the impact of a given verification mechanism to the ability to verify trustworthiness of membership views.

C. Membership view length

The first experiment is designed to investigate how the length of the membership view and position of the ego vehicle in the view affects the number of solutions that matches the information currently known by the ego vehicle. Fig. 4 shows the number of solutions for five different view lengths and all the possible positions of the ego vehicle for each view length. The y-axis shows the number of distinct solutions in each case.

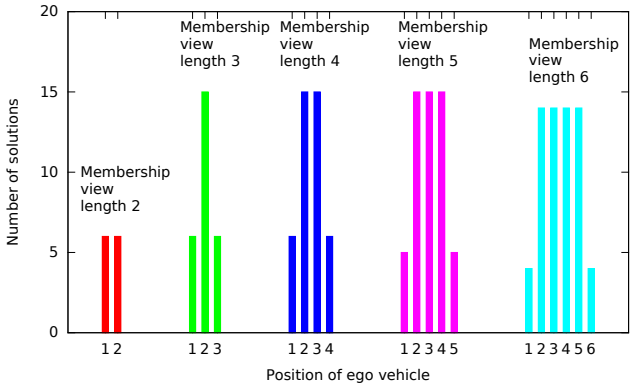

Fig. 4. The number of matching solutions for various platoon membership view lengths and position of the ego vehicle within those views.

The first thing to note in this figure is that if a vehicle is the head or tail of a membership view, then there are fewer solutions that would match what is currently known by the ego vehicle. This can be directly attributed to our assumption that a vehicle knows when it is at the head or tail of a platoon by the virtue of its on-board sensors.

The second phenomena that can be seen in the figure is that longer membership views correspond to slightly fewer solutions. While this might first seem counter-intuitive it can be explained by one of the experiment parameters which limit the number of faulty vehicles to 2. A longer false membership view requires more vehicles to be faulty, so there are fewer solutions when there are at most 2 faulty vehicles.

Note that even if the ego vehicle (which is assumed to be correct) is the platoon leader, there is still a possibility of erroneous membership views. This can happen since the platoon leader needs information from the platoon members to form a membership view. If the platoon leader receives incorrect information, the resulting membership view can also become incorrect even if the platoon leader is correct.

D. Verification mechanisms

One of the main research questions that motivates this work is to systematically find out how powerful different verification mechanisms are in terms of avoiding false membership views. We designed an experiment with six distinct verification mechanisms as described in Section III. We then ran the model with all $2^6 = 64$ possible combinations of mechanisms and logged the number of solutions possible for each combination. Each of the mechanisms can be shown to have an effect in at least some combination with other mechanisms, but some are stronger than others.

Fig. 5 shows the Shapley value for each of the six verification mechanisms for the scenario of a membership view of 3 vehicles and the ego vehicles is in second position. This particular choice of parameters is chosen since it maximises the potential number of solutions (cf. the first configuration with 15 solutions in Fig. 4). The sum of all Shapley values in this case is 12 which corresponds to the number of solutions that can be ruled out given that all verification mechanisms are activated (there are 3 solutions that cannot be ruled out, $15 - 3 = 12$).

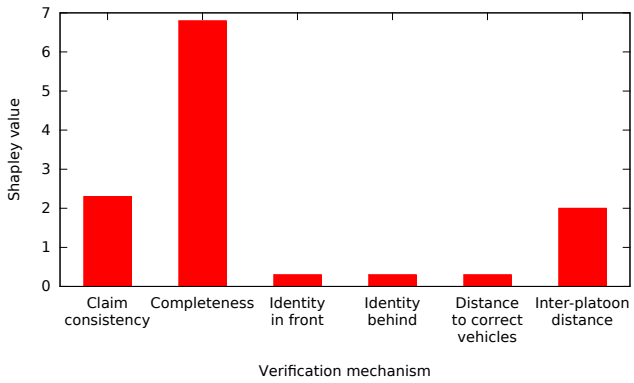

Fig. 5. The relative strength of the different verification mechanisms.

The first thing to note is that all verification mechanisms have a non-zero Shapley value which means that there is at least one configuration where that particular detection capability has an impact. Secondly, we see that there are three mechanisms that are significantly more powerful than the three least effective mechanisms. Starting with the most powerful capability, the requirement that claims are received from all members in a membership view clearly rules out many solutions in which the view can be false. Surprisingly, this capability is even more powerful than the ability with second highest Shapley value, which is to cross-check all the received claims and membership views to make sure that there are no contradictions. The third most powerful capability in this ordering is the ability to verify distances to other platoons in the system which can be used to rule out two platoons that drive so close as to be a single formation.

The other three verification mechanisms all received a Shapley of 0.375 for this particular parameter setting. This seems to indicate that they are less effective methods in terms of pure number of solutions that can be ruled out. However, further analysis is probably needed, since we have not considered the fact that some solutions could potentially be more dangerous than others.

E. Fault cardinality

In a cooperation where the majority of the involved members are faulty, it can be difficult to ensure information accuracy. This holds true also for platoon membership views. The final experiment is intended to find out exactly how the number of attackers impact the ability of an correct vehicle to have confidence in its membership view. In Fig. 6 the number of possible physical solutions for a given membership view is shown for varying number of attackers and verification mechanisms. The red bars in the figure represent the number of solutions when only the basic verification mechanisms are available, and the green bars correspond to the case when the vehicle has all of the verification mechanisms from Section III.

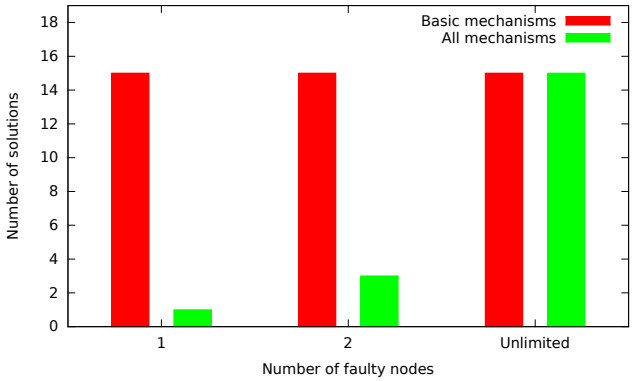

Fig. 6. The number of matching solutions for varying number of faulty vehicles.

If only the basic verification mechanisms are available, even a single attacker can corrupt the membership view of another vehicle. In contrast, if all the detection mechanisms are available then a single attacker cannot impose a false membership view on the other vehicles in the platoon (this is represented in the figure by the single solution indicated by the first green bar). With two colluding attackers, there are some cases of false membership views that cannot be detected.

Somewhat unexpectedly, if the number of attackers is unlimited, then there are just as many false membership solutions possible with and without verification mechanisms. Another way to put it is that with enough colluding attackers, there is no way to protect oneself against false membership information.

There are two interesting questions that arise from this negative result. The first question is how likely is a scenario where there are more than two attacking entities that try to disrupt platoon membership views? The second questions is whether the detection mechanisms discussed in this paper are really worthless in such a case, or whether there is a way forward even in this worst-case scenario. We will not try to answer the first question in this paper, as it is outside our current scope, but we consider it as an interesting topic for further work. With regard to the second question, there are some insights that can be gained by considering the number of microscopic solution metric discussed above. Given this way of counting solutions is shown in Fig. 7.

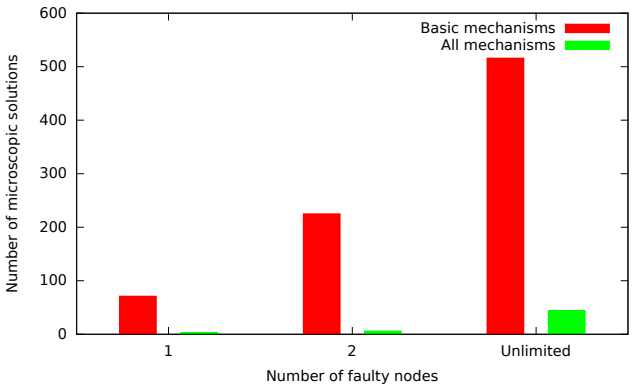

Fig. 7. The number of matching microscopic solutions for varying number of faulty vehicles.

Obviously, this way of counting results in a significantly higher number of solutions. The reason we are interested in this (otherwise slightly esoteric) metric is the case of unlimited number of attackers. Whereas in Fig. 6 there was no difference between having no extra verification mechanisms, and having the full set, Fig. 7 shows a drastic difference. This tells us that even if a vehicle might not be able to rule out false membership view in the case of multiple colluding attackers, it seems realistic that additional verification mechanisms could help.

VI. Related work

There is a wealth of research related to security in vehicular networks, see [9] for a recent survey. Given the legacy from research on MANETs many of the earlier works focused on security issues with regards to routing [15]. As the technology of inter-vehicle communication matured and became standardised, the focus shifted to warning systems and how to prevent spreading of false information [5].

There are several works that propose security measures based on some kind of data-verification. Dietzel et al. [8] use clustering to filter out false information in an aggregation-based protocol for information on vehicle speed in an area. Jaeger et al. [14] use Kalman filters to predict mobility movements of surrounding vehicles and compare that with the actually received information. This allows the system to identify non-plausible vehicle movements. Generally, verification can only be done if there are two independent information sources that can be compared. Aslam et al. [2] explore mechanisms to forward data using independent groups of vehicles, either by separation in space or by separation in time.

Our work touches on the problem of how to achieve agreement between cooperating vehicles in presence of failures (e.g., [11]). A key difference is that we are mainly concerned with how to determine the appropriate group (that correspond to physical reality) in which consensus can be reached.

Security in the context of vehicular platoons have been investigated by Studer et al. [23] who employ a combination of ensuring validity of data over time to verify that a vehicle is

travelling in the same convoy, and distance-based verification using time-of-flight of messages and MAC-layer timestamps. Lyamin et al. [16] present an algorithm for detecting jamming of messages in a platoon. Papadimitratos et al. [19] investigate the cost of security mechanisms on the communication performance and the implications for safety in an extreme platoon scenario with 100 vehicles.

Compared to the above approaches, our work can be seen as largely orthogonal. Our focus is not to provide a mechanism for intrusion detection, but provide the possibility to reason about such mechanisms in a formal manner. As a preliminary step in our analysis of security of platoon membership we performed manual analysis on some security measures [3].

VII. Conclusion and future work

We have presented a model-based approach for reasoning about correctness of platoon membership views. We presented a basic model with the bare minimum in terms of cooperative awareness claims, membership views and physical positions of vehicles. Still, this model allows us to automatically generate attack scenarios in a manner that would require significant manual effort even for the basic cases.

In our evaluation of how effective various verification mechanisms are in ruling out the possibility of scenarios with erroneous membership information, there were two unexpected outcomes. First, while we expected that having complete information from all platoon members would be important, the fact that it was much more important than actually checking claim consistency was a surprise. Second, we expected the ability to verify the identity of vehicles in front or behind to be more important than the quantitative analysis suggests.

While we believe the approach presented in this paper could potentially be valuable as a component in a security framework (e.g., [21]), there are some limitations that warrant further investigation. Obviously, the information being exchanged between the vehicles can be extended with more details about position, speed and various other sensor data readings. Moreover, while our framework allows a vehicle to explicitly trust one or more neighbouring vehicles, it would be interesting to allow multiple trust levels [17] that can be based on historical interactions.

Our assessment of verification mechanisms in this paper is purely quantitative in terms of counting how many scenarios with erroneous membership views that can be ruled out. It would be interesting to rank these scenarios according to severity and likelihood. Having 10 possible scenarios that are all low-risk scenarios would probably be preferable to having 3 possible scenarios associated with high risk.

In ongoing work we are integrating our framework as a component in a platoon membership protocol to assess its behaviour in a simulation environment with varying traffic conditions.

VIII. Acknowledgement

This work was supported by Centrum för industriell informationsteknologi (CENIIT), project 14.04. Special thanks also to Henrik X. Petterson at from Scania CV AB for his valuable input to this work.

References

[1] M. Althoff, O. Stursberg, and M. Buss. Model-based probabilistic collision detection in autonomous driving. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2), 2009, doi: 10.1109/TITS.2009.2018966.

[2] B. Aslam, S. Park, C. C. Zou, and D. Turgut. Secure traffic data propagation in vehicular ad hoc networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 6(1), 2010, doi: 10.1504/IJAHUC.2010.033823.

[3] M. Asplund. Poster: Securing vehicular platoon membership. In *Vehicular Networking Conference (VNC), 2014 IEEE*, 2014, doi: 10.1109/VNC.2014.7013324.

[4] AutoNet2030. European fp7 project co-operative systems in support of networked automated driving by 2030. http://www.autonet2030.eu/.

[5] N. Bissmeyer, K. Schroder, J. Petit, S. Mauthofer, and K. Bayarou. Short paper: Experimental analysis of misbehavior detection and prevention in vanets. In *Fifth IEEE Vehicular Networking Conference, VNC 2013*, pages 198–201. IEEE Communications Society, 2013.

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 6–6. USENIX Association, 2011.

[7] G. V. Chockler, I. Keidar, and R. Vitenberg. Group communication specifications: A comprehensive study. *ACM Comput. Surv.*, 33(4), 2001, doi: 10.1145/503112.503113.

[8] S. Dietzel, J. Gurtler, R. van der Heijden, and F. Kargl. Redundancy-based statistical analysis for insider attack detection in vanet aggregation schemes. In *Vehicular Networking Conference (VNC), 2014 IEEE*, 2014, doi: 10.1109/VNC.2014.7013332.

[9] R. G. Engoulou, M. Bellache, S. Pierre, and A. Quintero. VANET security surveys. *Computer Communications*, 44(0), 2014, doi: 10.1016/j.comcom.2014.02.020.

[10] European Telecommunications Standards Institute (ETSI). Intelligent Transport Systems (ITS). http://www.etsi.org/index.php/technologies-clusters/technologies/intelligent-transport.

[11] N. Fathollahnejad, E. Villani, R. Pathan, R. Barbosa, and J. Karlsson. On probabilistic analysis of disagreement in synchronous consensus protocols. In *Dependable Computing Conference (EDCC), 2014 Tenth European*, 2014, doi: 10.1109/EDCC.2014.26.

[12] G. Guette and C. Bryce. Using tpms to secure vehicular ad-hoc networks (vanets). In J. Onieva, D. Sauveron, S. Chaumette, D. Gollmann, and K. Markantonakis, editors, *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, volume 5019 of *Lecture Notes in Computer Science*, pages 106–116. Springer Berlin Heidelberg, 2008, doi: 10.1007/978-3-540-79966-5˙8.

[13] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7), 2000, doi: 10.1109/5.871301.

[14] A. Jaeger, N. Bimeyer, H. Stbing, and S. Huss. A novel framework for efficient mobility data verification in vehicular ad-hoc networks. *International Journal of Intelligent Transportation Systems Research*, 10(1), 2012, doi: 10.1007/s13177-011-0038-9.

[15] T. Leinmuller, E. Schoch, and F. Kargl. Position verification approaches for vehicular ad hoc networks. *Wireless Communications, IEEE*, 13(5), 2006, doi: 10.1109/WC-M.2006.250353.

[16] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo. Real-time detection of denial-of-service attacks in ieee 802.11p vehicular networks. *Communications Letters, IEEE*, 18(1), 2014, doi: 10.1109/LCOMM.2013.102213.132056.

[17] R. Machado and K. Venkatasubramanian. Short paper: Establishing trust in a vehicular network. In *Vehicular Networking Conference (VNC), 2013 IEEE*, 2013, doi: 10.1109/VNC.2013.6737611.

[18] D. Nicol, W. Sanders, and K. Trivedi. Model-based evaluation: from dependability to security. *Dependable and Secure Computing, IEEE Transactions on*, 1(1), 2004, doi: 10.1109/TDSC.2004.11.

[19] P. Papadimitratos, G. Calandriello, J.-P. Hubaux, and A. Lioy. Impact of vehicular communications security on transportation safety. In *INFOCOM Workshops 2008, IEEE*, 2008, doi: 10.1109/INFOCOM.2008.4544663.

[20] J. Scahill and J. Begley. The great sim heist: How spies stole the keys to the encryption castle. *The Intercept*, 2015.

[21] R. K. Schmidt, T. Leinmüller, E. Schoch, A. Held, and G. Schäfer. Vehicle behavior analysis to enhance security in vanets. In *Proceedings of the 4th IEEE Vehicle-to-Vehicle Communications Workshop (V2VCOM2008)*, 2008.

[22] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. Lo Cigno. Supporting platooning maneuvers through IVC: An initial protocol analysis for the join maneuver. In *Wireless On-demand Network Systems and Services (WONS)*, 2014, doi: 10.1109/WONS.2014.6814733.

[23] A. Studer, M. Luk, and A. Perrig. Efficient mechanisms to provide convoy member and vehicle sequence authentication in vanets. In *Third International Conference on Security and Privacy in Communications Networks (SecureComm)*, 2007, doi: 10.1109/SECCOM.2007.4550363.

[24] B. Yu, C.-Z. Xu, and B. Xiao. Detecting sybil attacks in VANETs. *Journal of Parallel and Distributed Computing*, 73(6), 2013, doi: 10.1016/j.jpdc.2013.02.001.