Securing Cooperative Intersection Management through Subjective Trust Networks

Frank Kargl, Nataša Trkulja, Artur Hermann
Institute of Distributed Systems
University of Ulm
Ulm, Germany
{frank.kargl, natasa.trkulja, artur.hermann@uni-ulm.de}

Florian Sommer
Institute of Energy Efficient Mobility
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
florian.sommer@h-ka.de

Anderson Ramon Ferraz de Lucena, Alexander Kiening
DENSO AUTOMOTIVE Deutschland GmbH
Eching, Germany
{a.ferraz, a.kiening@eu.denso.com}

Sergej Japs
Product Engineering
Fraunhofer IEM
Paderborn, Germany
sergej.japs@iem.fraunhofer.de

***Abstract*—Connected, cooperative, and autonomous mobility (CCAM) will take intelligent transportation to a new level of complexity. CCAM systems can be thought of as complex Systems-of-Systems (SoSs). They pose new challenges to security as consequences of vulnerabilities or attacks become much harder to assess. In this paper, we propose the use of a specific type of a trust model, called subjective trust network, to model and assess trustworthiness of data and nodes in an automotive SoS. Given the complexity of the topic, we illustrate the application of subjective trust networks on a specific example, namely Cooperative Intersection Management (CIM). To this end, we introduce the CIM use-case and show how it can be modelled as a subjective trust network. We then analyze how such trust models can be useful both for design time and run-time analysis, and how they would allow us a more precise quantitative assessment of trust in automotive SoS. Finally, we also discuss the open research problems and practical challenges that need to be addressed before such trust models can be applied in practice.**

***Index Terms*—Automotive security, system-of-systems security, automotive trust models**

I. Introduction

The development of vehicles and their surrounding infrastructure leads to ever more complex systems. A recent culmination of this trend are connected, cooperative, and autonomous mobility (CCAM) systems that will enable applications like cooperative intersection management (CIM) in which vehicles cooperate with Vehicular Edge Computing devices to efficiently manage traffic at an intersection. Considering also the internal complexity of the vehicles themselves, we end up with extremely complex systems-of-systems.

The aforementioned CIM and many other CCAM applications assume that the vehicles provide correct information and – where intended – follow the decisions of roadside infrastructure such as the multi-access edge computing (MEC) servers co-located with roadside units (RSUs). This assumption may not always be true due to various reasons [1]. For example, a vehicle with a compromised communication unit could send an incorrect position to the MEC server claiming that the vehicle is entering the intersection from a different direction. In the worst case, this could affect vehicles' safety and lead to accidents. This is why security mechanisms like misbehavior detection [2] for connected vehicles and cooperative mobility are important and have been under investigation for many years.

However, the trend towards ever more complex automotive SoS makes design and deployment of appropriate security controls more and more challenging. Security engineering as defined by standards like ISO/SAE 21434 and mandated by UNECE R 155 regulation has become commonplace in the automotive industry. However, such security engineering has two drawbacks: First, its analysis often focuses only on single vehicles or even only on sub-components, leaving the overall CCAM SoS out of scope. Second, its approaches are limited in their accuracy as they are not based on accurate mathematical modeling of the systems and their inter-dependencies.

It might, for example, become very difficult to assess how a vulnerability found in one particular Electronic Control Unit (ECU) of a vehicle might affect security and thus safety of the overall CIM application. For traditional risk assessment conducted during a security engineering process, assessing how a vulnerability transitively affects other components in a complex system-of-systems is not well defined.

In this paper, we therefore propose a trust management framework that allows to mathematically model the trust relationships between components in an automotive system-of-systems as a trust graph or trust network. Using this framework, questions as the one described above could be answered in a quantitative manner: how is, for example, the trustworthiness of a cooperative intersection application

This work was partially supported by the SecForCARs-SAVE project (reference number: 16KIS0796) which is funded by the Federal Ministry of Education and Research of Germany (BMBF).
This work is also partially funded by the European Union under grant agreement no. 101069688. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

affected by a vulnerability in one of the participating vehicles? Are the available information and the driving commands derived thereupon in a CIM-managed intersection trustworthy enough to continue operation? Or would one have to resort to a fail-operational state where vehicles do not rely on the cooperation with potentially untrustworthy vehicles? Our trust management framework build on the formal logic framework of Subjective Logic [3], which is a well-established formalism to reason about trust in systems but has not yet been used to model automotive CCAM SoS.

In the remainder of this paper, we first introduce an example scenario by which we will:
1. illustrate an example scenario for CIM,
2. identify challenges and requirements for a trust model in CIM and other CCAM systems,
2. review related work to identify candidates for the trust model and related solutions,
3. introduce an approach to design a trust framework for our CIM example, and
4. show how our trust framework can be used to evaluate whether the complex CIM application is operating in a trustworthy state.

With the use-case driven approach taken here, we want to demonstrate the feasibility of our trust framework and at the same time identify the open challenges and research questions that still need to be addressed in order to design and build a generalisable framework.

II. Cooperative Intersection Management (CIM)

CIM is an approach that uses communication and cooperation between vehicles and infrastructure, or just between vehicles, to decide when which vehicle should pass through the intersection. In this way, safety and efficiency at intersections can be improved.

Approaches for a CIM can be divided into a signalized and a non-signalized CIM. In a signalized CIM, there are traffic signals at the intersection, and traffic passes through the intersection according to the traffic signals. This is not the case in a non-signalized CIM where the traffic control must be done by other means, such as by a central authority that tells each vehicle when it can pass through the intersection. CIM can be further categorized into centralized or distributed CIM. Centralized CIM relies on a coordination unit, such as a MEC server, that collects information from the vehicles and tells the vehicles when and how they can pass through the intersection. In distributed CIM, there is no such central entity. Instead, vehicles communicate with each other and take their decisions locally [1].

In this paper, we focus on non-signalized and centralized CIM, which is a reasonably complex and representative application for a CCAM scenario. We argue that other CIM approaches or many CCAM applications are similar in nature, system structure, and complexity and, thus, similar trust models can be used there as well.

There are several approaches to implement a non-signalized and centralized CIM. A common approach is that each vehicle driving towards the intersection reserves a trajectory for a limited time window during which it passes the intersection (see Figure 1). For this purpose, vehicles send several attributes to the MEC server such as their turn direction, position, and speed. Based on these attributes, the MEC server can calculate when the vehicle will arrive at the intersection and which trajectory it will use. The MEC server then reserves a time window for the vehicle to pass the intersection and sends it back to the corresponding vehicle [4]. The attributes that are exchanged between the vehicles and the MEC server are summarized in Figure 2.

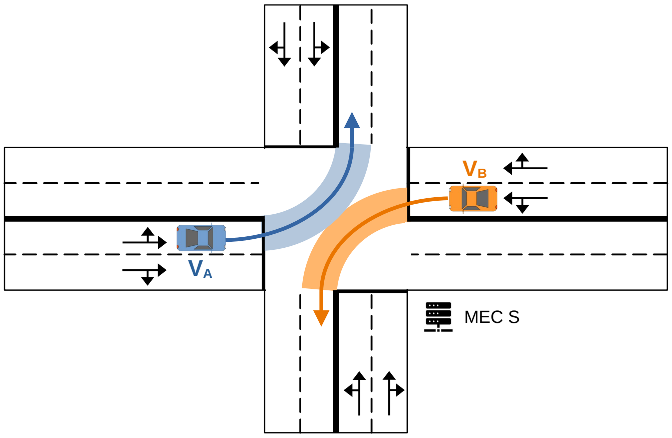
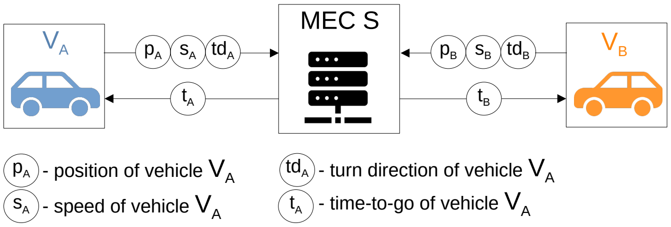


Fig. 1. Visualization of the reservation of trajectories for the individual vehicles (blue and orange areas)



V_A
p_A
s_A
td_A
t_A
MEC S
p_B
s_B
td_B
t_B
V_B
p_A - position of vehicle V_A
s_A - speed of vehicle V_A
td_A - turn direction of vehicle V_A
t_A - time-to-go of vehicle V_A

Fig. 2. Overview of the information exchange between the vehicles and the MEC server

III. Requirements and Challenges for CCAM Trust Models

After introducing the CIM scenario, we can now specify the requirements but also identify the challenges when trying to derive a matching trust model that reflects all trust relationships in the scenario.

One requirement is that the trust model should *allow transitive trust relationships*. In a system-of-systems, transitive information flows exist across multiple nodes. Each node involved in the data flow has an impact on the trustworthiness of the data, which is taken into account in the trust calculation through transitive trust relationships.

Regarding trust calculation, the trust model should allow to *reflect trust in data and trust in nodes*, since we are ultimately interested in trust in data, but also need to consider trust in nodes to transitively reflect trust of data relayed by nodes.

The trust reasoning mechanism should be *probabilistic and support the expression of uncertainty*. Probabilistic reasoning is important since trust in another entity is not binary, but should reflect a continuum from not trusted to fully trusted. In addition, the trust opinion should include uncertainty, since in some cases a node may not have enough evidence to establish reliably with what probability a node is trustworthy.

Furthermore, we need to represent *subjective notions of trust*, as each vehicle will be presented with different evidence. Therefore, it is important to express trust from the perspective of and based on evidence that a single node has.

The entire process of calculating trust opinions should be fast and efficient so that it can be *used at run-time for real-time applications* even on resource-constraint devices. At the same time, it can also be *used during design time*, for example, in the context of risk management where the trust model could be used to better quantify risks in a complex automotive SoS.

Finally, the trust model should be generic enough to be applicable in different CCAM systems and not only in CIM scenarios. An approach that meets all of these requirements is described in this paper.

IV. Related work

Security issues in CCAM systems can result in safety problems and is therefore of highest importance. As this was evident even when starting to design vehicle-to-everything (V2X) systems, security mechanisms have already been investigated and introduced from the start on. In this section, we describe some of these works that are relevant also for trust modeling.

A. Misbehavior Detection

Misbehavior detection (MBD) was originally developed in the context of V2X communication and can be divided into node-centric and data-centric misbehavior detection. Van der Heijden et al. [2] provide an overview over the different forms of MBD. They distinguish between node-centric misbehavior detection which examines the behavior of nodes, like vehicles or RSUs, to detect misbehavior. In contrast, data-centric misbehavior detection examines the information that is communicated to detect misbehavior, regardless of who transmitted the message. For example, Ruj et al. [5] detect misbehaving nodes in that a vehicle, after receiving a message from another vehicle, compares the position indicated in the message with the estimated position of the sending vehicle. Many authors propose combinations of both types leading to trust-based approaches that use outputs of data-centric and node-centric misbehavior detection mechanisms to form trust opinions on other vehicles and the data they send.

B. Trust Management Frameworks

Garlichs et al. [6] have proposed a trust management framework that is to be used in a vehicle platoon system. Depending on how much a host vehicle trusts its predecessor in the platoon, the host vehicle regulates its safety distance to that vehicle. The trust value is derived by the host vehicle comparing the actual behavior of the sender with the information it has received from the sender over an extended period of time. The calculated trust value is compared with predefined values to decide which safety distance to use. While shown to be effective and efficient, the mechanism is highly application specific and provides a rather static trust model.

Another approach was proposed by Raya et al. [7]. Here, vehicles detect the misbehavior of other vehicles through various MBD mechanisms and based on this, exchange accusations about potentially malicious vehicles. If the sum of weighted accusations against a vehicle exceeds a certain threshold, the trust of the corresponding vehicle is too low so that it is temporarily removed from the network.

In some related work, more advanced forms of decision logic are used, such as in Raya et al. [8]. Here, several node-centric and data-centric attributes are used to calculate the trust value of a node. Based on this value, it is decided whether the respective node should be trusted or not. For this purpose, several decision logics were tried out and compared, such as weighted voting and Bayesian inference. However, no mechanism performed best in all simulated scenarios.

Dietzel et al. [9] and van der Heijden et al. [10] were the first to propose the use of subjective logic to merge results from different detection mechanisms in a misbehavior detection system through the use of Subjective Logic. Their goal was to integrate an arbitrary number of MBD mechanisms into a single framework to enhance detection accuracy. Müller et al. [11] have likewise used subjective logic to generate trust opinions on nodes for MBD. Based on consistency checks of the messages sent by the nodes, inconsistent messages are detected, upon which the trust of the corresponding nodes is adjusted. In this way, misbehaving nodes could efficiently be detected and isolated.

Sohail et al. [12] use subjective logic to improve a distance vector routing protocol used in V2X networks. Generated trust opinions were included in the trust fields of the routing table, making the protocol more robust against malicious vehicles that could, for example, drop the messages. Müller et al. [13] use subjective logic to determine the reliability of data provided by road side units (RSUs) to vehicles.

Beyond the context of V2X networks, trust management frameworks were also developed. For example, Kurdi et al. [14] have proposed a trust management framework for cloud service providers (CSPs) and Dimitrakos et al. [15] have described a trust-based authorization approach for the Internet of Things. Here, a trust value is calculated based on subjective logic integrating several information sources. Based on this trust value, it is decided whether an entity is authorized to access resources or not.

The discussion in this section shows that the use of trust management frameworks in the automotive domain is mostly limited to misbehavior detection only. But as the last two related works show, trust management can be considered for much broader applications than just misbehavior detection.

Therefore, we go beyond misbehavior detection and suggest a more general trust management framework for reasoning about trust in complex system-of-systems in the automotive domain.

As subjective logic has proven to be highly useful for such a task, we will likewise base our framework on subjective logic, since it fulfills all the previously mentioned requirements.

V. Trust Management Framework

Our trust management framework means to model trust relationships in cyber-physical systems like CCAM where nodes exchange messages and information that affect the behavior of other vehicles, thus, forming a cooperative system. In such systems, nodes need to put some degree of reliance on or trust in the correctness of received information, as incorrect information could negatively affect their own behavior or also information they send onward. To avoid blind trust, such functional trust, i.e., trust in the correct execution of a specific function or service between entities, should be established by explicit mechanisms. Representing the structure of such trust relationships and quantifying its degree by collecting evidence is the purpose of the trust management framework.

For example, in our Cooperative Intersection Management use-case, vehicles would send Cooperative Awareness Messages (CAM)[1] to other vehicles and MEC servers informing them about their mutual position, speed, and turn direction. Once the receiving vehicle process the CAM and detect risks for collisions in the intersection, they may decide to reduce their speed or stop before entering the intersection to avoid potential accidents. Likewise, the MEC server will base its decision of intersection time slot allocation for each vehicle on such information. Therefore, the messages sent by vehicles or the MEC server will clearly affect the behavior of the other vehicles, rendering message integrity and correctness of contained information a highly important aspect in CCAM. Therefore, it is crucial to establish a way to continuously evaluate whether any node in a CCAM system can trust the data it received and the nodes it received the data from.

To this end, the trust management framework presented here will enable every node in the CCAM system to decide whether nodes and/or their data can be trusted. Our trust management framework is based on the concepts of subjective logic and will allow each node X (trustor) to form a subjective opinion, $\omega_y^X = (b_y^X, d_y^X, u_y^X)$, on the trustworthiness of the data y sent by node Y (trustee). Here, b_y^X is the belief that X has in y, d_y^X is the disbelief that X has in y, and, finally, u_y^X is the amount of uncertainty X has w.r.t. the trustworthiness of y. Like this data-centric trust, our framework also supports node-centric trust opinions ω_Y^X that express the functional trust that X has in Y to provide correct service, e.g., to produce and send correct data. When X uses some data in its actual driving functions, we call ω_y^X the **actual trust level (ATL)**.

Our framework follows a *zero-trust* principle, meaning that a trustor X initially has a zero belief ($b_y^X = 0$) and an uncertainty of 1 ($u_y^X = 1$) towards data y or trustees Y. Once more evidence is collected about the trustworthiness of y or Y, for example through misbehavior detection mechanisms, the belief will be recalculated. The belief will also continue to be updated on a regular basis as new information or evidence arrives in the system.

For deciding whether a certain ATL is sufficiently high for a certain CCAM function, the trustor node X will need to compare the ATL of the data y to the **required trust level (RTL)**. The RTL is also an opinion, $\omega_y^R = (b_y^R, d_y^R, u_y^R)$, but, unlike ATL, its values are typically established during system design-time and reflect the degree of negative impact that reliance on malicious input could have. The process of building ATL and RTL is discussed in more detailed in the following sections.

Comparing ATL and RTL opinions can be done based on their projected probabilities as calculated using the following equations [3]:

$$P_i^j = b_i^j + a_i^j \times u_i^j \tag{1}$$

where a_i^j is the *base rate* of i, i.e., the prior probability in absence of specific evidence. Then P_y^X must exceed P_y^R for y to be considered trustworthy by node X:

$$P_y^X > P_y^R \tag{2}$$

Arguably, comparing projected probabilities of the ATL and the RTL is a simplistic approach to determine the trustworthiness of a node. Further investigation into more sophisticated comparisons will be done in future work that could also involve a more refined consideration of the degrees of uncertainty of the two opinions.

A. Actual Trust Level (ATL)

A node X in a CCAM system that wants to determine the actual trust level (ATL) of data y sent from node Y has to perform the following three steps (see also Figure 3):

1) Form an opinion of the node Y based on some Trustworthiness Indicators (TI) for Y (ω_Y^X) - **node-based trust**.
2) Perform trust-discounting between the opinion it has on the node Y (ω_Y^X) and the opinion that the node Y has on its own data (ω_y^Y) to obtain the opinion on the data y it received (ω_y^X) - **data-based trust**.
3) Create additional opinions on data y based on own assessments of the trustworthiness of y, then fuse all these opinions with (ω_y^X).

For this, a node Y that sends data y to another node X also has to include a trust opinion that it has formed on the trustworthiness of its own data, ω_y^Y, in the same message.

To illustrate how this three-step process works, we will use our example of Cooperative Intersection Management (CIM). As shown in Figure 2, once vehicles V_A and V_B have approached an intersection managed by a MEC S, they send various attributes to the MEC. For the sake of simplicity of this example, we focus only on position data p_A and p_B. The vehicles include their own opinions on their position data,

[1] in IEEE 1609, this corresponds to a Basic Safety Message (BSM)

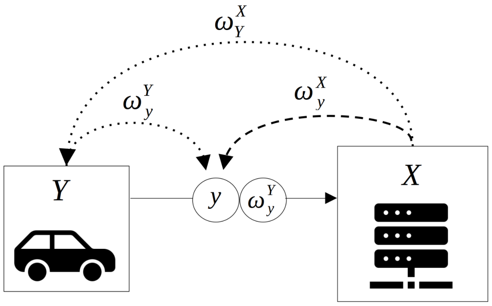


Fig. 3. Forming of trust opinions

$\omega_{p_A}^{V_A}$ and $\omega_{p_B}^{V_B}$ respectively, to the MEC (see Figure 4). Such opinions could be formed by A and B based on, for example, the knowledge about GPS reception accuracy or other factors.

Once S receives the position data and the opinions, it first computes the trust opinions, $\omega_{V_A}^S$ and $\omega_{V_B}^S$, on the vehicles V_A and V_B (**node-based trust**). It does this based on a set of *trustworthiness indicators* (TIs). Trustworthiness indicators serve to establish node-centric trust into another node. They will either increase the belief, the disbelief or the uncertainty of the opinion that the MEC has on a vehicle. For example, if the remote vehicle has a valid certificate, this will increase trust in it. If a TI like a certificate cannot be evaluated, uncertainty will raise. And if the node has shown earlier misbehavior, node reputation will decrease and disbelief will increase. Future work needs to establish a more detailed understanding of trust indicators for node-centric trust, but literature knows a vast amount of mechanisms that this can be built on [2].

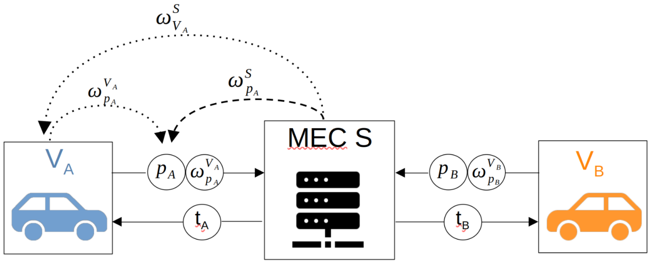


Fig. 4. Trust model applied to CIM

After the MEC S has calculated trust opinions on the vehicles themselves, it proceeds to compute the opinions on the position data p_A and p_B sent by the vehicles (**data-based trust**). It does this through performing trust discounting between: i) the opinion it formed on the vehicles $\omega_{V_A}^S$ and $\omega_{V_B}^S$, and ii) the opinions of the vehicles have on their own position data $\omega_{p_A}^{V_A}$ and $\omega_{p_B}^{V_B}$. This is calculated as follows [3]:

$$\omega_{p_A}^{[S;V_A]} := \begin{cases} b_{p_A}^{[S;V_A]}(x) & = P_{V_A}^S * b_{p_A}^{V_A}(x) \\ u_{p_A}^{[S;V_A]} & = 1 - P_{V_A}^S * \sum_{x\in X} b_{p_A}^{V_A}(x) \\ a_{p_A}^{[S;V_A]}(x) & = a_{p_A}^{V_A}(x) \end{cases} \tag{3}$$

$$\omega_{p_B}^{[S;V_B]} := \begin{cases} b_{p_B}^{[S;V_B]}(x) & = P_{V_B}^S * b_{p_B}^{V_B}(x) \\ u_{p_B}^{[S;V_B]} & = 1 - P_{V_B}^S * \sum_{x\in X} b_{p_B}^{V_B}(x) \\ a_{p_B}^{[S;V_B]}(x) & = a_{p_B}^{V_B}(x) \end{cases} \tag{4}$$

The MEC now has a single opinion on the position data that it received from the vehicles and it could assign this value to the ATL, calculate the projected probability, and compare it with the projected probability of the pre-established RTL to decide whether the position data is trustworthy. However, for a more robust and a more accurate calculation of trustworthiness, the MEC should collect additional opinions on the position data from other sources and fuse all of these opinions together. For example, the MEC can be equipped with camera-based sensors that independently determine the positions of each vehicle. The MEC can then calculate additional opinions on the position data reported by the vehicles by comparing them to the position data determined by the cameras. These opinions would all be fused together to create one final set of opinions on the position data p_A and p_B. We discuss this in our CIM example from Figure 4 with concrete numerical values for the opinions.

Here we look only at vehicle V_A and MEC server S exchanging data and trust opinions. We start by assuming that the S has been observing vehicle V_A for a while through a reputation system. Such a reputation system can be based on average data-centric trust in data sent by the transmitting vehicle. Based on this reputation system for V_A, S has developed an opinion on the trustworthiness of the vehicle, $\omega_{V_A}^S = (0.9, 0.05, 0.05, 0.9)$. Such an opinion expresses a high level of trust given that the belief value is 0.9 and it also expresses a high certainty in this judgement as the uncertainty is relatively low 0.05.

Moreover, the vehicle V_A is absolutely convinced that its position data is sound, so the opinion it has on its own data is $\omega_{p_A}^{V_A} = (1.0, 0.0, 0.0, 0.5)$, which expresses complete trust as the belief is equal to 1. In realistic settings, GPS and sensor inaccuracies could be accounted for by putting less belief and more uncertainty into this opinion.

Given that we now have both the MEC's opinion on the vehicle, $\omega_{V_A}^S$, as well as the opinion of the vehicle on its data, $\omega_{p_A}^{V_A}$, we can now use trust discounting as in Formula 3 to calculate the first opinion which S has on the data it received, resulting in $\omega_{p_A}^{[S;V_A]} = \omega1_{p_A}^S = (0.95, 0, 0.05, 0.5)$.

Using a camera-based sensor that S is equipped with, S compares the position data reported by the vehicle and the position data produced by its camera to create a second opinion on the data received, $\omega2_{p_A}^S = (0.75, 0.1, 0.16, 0.75)$ which shows lower belief but also a higher degree of uncertainty.

Finally, S uses cumulative belief fusion [3, Definition 12.5] to fuse the two opinions into a joint opinion:

$$\omega_{p_A}^{1\diamond 2} = \omega1_{p_A}^S \bigoplus \omega2_{p_A}^S \tag{5}$$

Plugging the appropriate values into this equation, we obtain $\omega_{p_A}^{1\diamond 2} = (0.93, 0.03, 0.04, 0.56)$, as well as a projected probability of $p_{\omega_{p_A}^{1\diamond 2}} = 0.96$, which represents our ATL. In other words, our MEC server has established a very high belief in the trustworthiness of the position it received from V_A. This was to be expected given that it had a high belief into the trustworthiness of A, A had high trustworthiness into its position measurement, and S also used another camera-based sensor to provide additional position evidence. As shown, this subjective logic trust network can integrate various evidence sources in a complex trust network that can extend significantly beyond this simple example.

However, the ATL on its own does not allow for a decision to be made on whether the MEC server should use the position data as reported by vehicle V_A. For this, the ATL has to be evaluated in the context of a specific application like CIM and its trust requirements by being compared to an appropriate RTL.

B. Required Trust Level (RTL)

Typically, the RTL is established during CCAM system development and represents a fixed value expressing how critical the trustworthiness of some data is in the context of a specific driving function or application. In this section, we provide an approach to defining the RTL during a CCAM system development process. ISO/SAE 21434 defines a security-specific development process that must be fulfilled by vehicle manufacturers and suppliers. For this reason, it makes sense to determine the RTL based on activities during this process. In particular, we see three main activities as relevant: The *Threat Analysis and Risk Assessment (TARA)*, *derivation of a security concept and security mechanisms*, and the *verification and validation*. During TARA, the vehicle or its subsystems (so-called *Items*, e.g., control units) are examined for potential cyber-threats. The risk of identified threats is then assessed. This usually takes into account the feasibility of an attack and its impact on the system. Based on the TARA results, requirements for a cybersecurity concept are derived. This includes an implementation of security mechanisms (e.g., authentication of in-vehicle messages). In the subsequent verification and validation phase, implemented security solutions are evaluated. This can be done, for example, by testing the correct functionality of the security mechanisms or by carrying out penetration tests to identify potential remaining vulnerabilities.

In terms of the approach presented in this paper, these activities can be used to determine a RTL for the vehicle, individual vehicle components, but it could also be extended to complex CCAM systems. The threats and their risks identified during TARA have a significant influence on this value. For example, if only a few non-critical threats are identified for a control unit, the RTL may be low. If, on the other hand, a large number of highly critical threats to this control unit are identified, a high RTL is necessary. In this case, the security concept or implemented security mechanisms can be used to refine the RTL, since they lower the feasibility of an attack. This would result in a lower RTL. Additional testing activities can lower that value even further, since the resilience against attacks is further evaluated in this case.

Thus, the question arises how these activities can be transferred into the subjective logic model in order to calculate the RTL. In order to answer this question, we will only consider the TARA in this paper as an example. The result of a TARA usually consists of a number of threats to the vehicle and their risk. The risk of the individual threats can in turn be determined based on certain factors, such as likelihood or attack feasibility. In order to use such factors to determine opinions, Jøsang proposes the use of qualitative tables. One example is the comparison of likelihood levels and confidence levels [3, p. 49]. A value within this table can be interpreted as an opinion. This can be applied to the case of a TARA. An initial approach that can be used to define the RTL would be the use of the Cybersecurity Assurance Level (CAL), a classification scheme defined in the ISO/SAE 21434 and which is used to define the level of rigor required for a satisfactory development of a component, from the cybersecurity perspective. For this, a TARA shall be done and cybersecurity goals (CG) shall be defined for the component(s) under analysis. The next step is to specify a CAL for each selected CG. The TARA is used to understand threats and their attack feasibility. As shown in the Table I, we can combine the CAL with the attack feasibility and determine an RTL.

TABLE I
USING CAL LEVELS OF THE ISO/SAE 21434 IN COMBINATION WITH THE ATTACK FEASIBILITY OF TARA THREATS TO DETERMINE OPINIONS.

	Attack Feasibility			
	Very Low	Low	Medium	High
CAL1	1V	1L	1M	1H
CAL2	2V	2L	2M	2H
CAL3	3V	3L	3M	3H
CAL4	4V	4L	4M	4H

To compute an RTL for the vehicle's position sensor in the CIM example, we assume that a CAL3 resulted from an executed TARA. To determine opinions for individual threats, we use two attacks from the Automotive Attack Database (AAD) [16] and its associated attack taxonomy [17]. The two attacks (AAD ID: ID2019_Regulus_SSA1, ID2018_Zeng_MSA1) describe spoofing of GPS data and position information about the vehicle. For each attack, a *Common Vulnerability Scoring System (CVSS)* rating is available, together with a rating according to its sub-metric *Exploitability*. To rank the attacks in Table I for CAL3, we use the *Exploitability* metric. For the first attack, there is an *Exploitability* of 3.89, which is the highest possible value. Thus, the table value 3H can be assigned to it. The second attack consists of two attack steps, each with an *Exploitability* of 0.9 and 2.8. For the example here, we take the mean value of 1.85, which can be assigned to the table entry 3L. Using the approach suggested by Jøsang [3], two opinions can be derived from the table entries. For the first attack, the opinion is ${\omega_1}_{p_A}^{V_A} = (0.73, 0, 0.27, 0.5)$. For the second attack, the opinion

${\omega_2}_{p_A}^{V_A} = (0.23, 0.5, 0.27, 0.5)$. Applying the cumulative fusion to the two opinions yields $\omega_{p_A}^{V_A(1\diamond 2)} = (0.55, 0.29, 0.15, 0.56)$ and thus the projected probability $\omega_{p_A}^{V_A(1\diamond 2)} = 0.63$.

VI. Conclusion & Future Work

In this paper, we introduce the concept of using subjective logic and subjective trust networks to model trust relationships in complex systems-of-systems for CCAM applications. We illustrate how such a trust network can be built for our CCAM use-case and how a comparison of an actual trust level (ATL) with a required trust level (RTL) can help to take trust-related decisions in CCAM systems. Furthermore, we discuss what data sources can be used to establish both the ATL and the RTL.

Many of our discussions introduce concepts and raise questions for future research. Among others, we are working on designing trust networks for larger scenarios. We also plan to investigate how trust networks scale and how attacks affect the trust networks in different parts of the graph. Moreover, we are linking trust models to different information sources that provide us trust evidence. Among these are hardware security mechanisms, misbehavior detection mechanisms, and many more. The goal is to provide a highly accurate structural model of trust interrelations but also to quantify the available trust evidence in order to allow precise statements on trustworthiness. A challenge here is that all this evidence needs to be represented in form of subjective logic opinions in order to be incorporated. Such translations are dependent on the semantics of the evidence and therefore need to be established differently for different kinds of evidence. Currently, the RTL is established in a rather static fashion at design time only. We envision that a more dynamic RTL could be established that also takes into consideration not only the static system architecture but also the information about the current driving situation and the safety margins that a vehicle can have in it.

In the long-term, our research could pave the way to a novel approach how to assess and reason about a systems security and trustworthiness, linking safety and security much closer than it is done today. The trust management framework and the methodology to design and apply trust models could also be transferred to many other domains of cyber-physical systems and the Internet-of-Things, like smart homes, industry 4.0, and many more.

Acknowledgment

We thank the partners in both the German SecForCARS-SAVE and the European CONNECT project for their fruitful discussions on trust management.

References

[1] L. Chen and C. Englund, “Cooperative intersection management: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[2] 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, 2019.

[3] A. Jøsang, *Subjective Logic – A Formalism for Reasoning Under Uncertainty*. Springer Cham, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-42337-1

[4] T. Niels, N. Mitrovic, N. Dobrota, K. Bogenberger, A. Stevanovic, and R. Bertini, “Simulation-based evaluation of a new integrated intersection control scheme for connected automated vehicles and pedestrians,” *Transportation Research Record*, vol. 2674, no. 11, pp. 779–793, 2020. [Online]. Available: https://doi.org/10.1177/0361198120949531

[5] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, “On data-centric misbehavior detection in vanets,” in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, 2011, pp. 1–5.

[6] K. Garlichs, A. Willecke, M. Wegner, and L. C. Wolf, “Trip: Misbehavior detection for dynamic platoons using trust,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 455–460.

[7] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-p. Hubaux, “Eviction of misbehaving and faulty nodes in vehicular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1557–1568, 2007.

[8] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, “On data-centric trust establishment in ephemeral ad hoc networks,” in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, pp. 1238–1246.

[9] S. Dietzel, R. van der Heijden, H. Decke, and F. Kargl, “A flexible, subjective logic-based framework for misbehavior detection in v2v networks,” in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*. Sydney, Australia: IEEE, Jun. 2014, pp. 1–6.

[10] R. W. van der Heijden, A. al Momani, F. Kargl, and O. Abu-Sharkh, “Enhanced position verification for vanets using subjective logic,” in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. Montreal, Canada: IEEE, Sep. 2016, pp. 1–7.

[11] J. Müller, T. Meuser, R. Steinmetz, and M. Buchholz, “A trust management and misbehaviour detection mechanism for multi-agent systems and its application to intelligent transportation systems,” in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, 2019, pp. 325–331.

[12] M. Sohail, L. Wang, S. Jiang, S. Zaineldeen, and R. Umair Ashraf, “Multi-hop interpersonal trust assessment in vehicular ad hoc networks using threevalued subjective logic,” *IET Information Security*, vol. 13, 05 2019.

[13] J. Müller, M. Gabb, and M. Buchholz, “A subjective-logic-based reliability estimation mechanism for cooperative information with application to iv’s safety,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1940–1946.

[14] H. Kurdi, B. Alshayban, L. Altoaimy, and S. Alsalamah, “Trustyfeer: A subjective logic trust model for smart city peer-to-peer federated clouds,” *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–13, 02 2018.

[15] T. Dimitrakos, T. Dilshener, A. Kravtsov, A. La Marra, F. Martinelli, A. Rizos, A. Rosetti, and A. Saracino, “Trust aware continuous authorization for zero trust in consumer internet of things,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1801–1812.

[16] F. Sommer and J. Dürrwang, “Ieem-hska/aad: Automotive attack database (aad),” 2019. [Online]. Available: https://github.com/IEEM-HsKA/AAD

[17] F. Sommer, J. Dürrwang, and R. Kriesten, “Survey and classification of automotive security attacks,” *Information*, vol. 10, no. 4, p. 148, 2019.