WIP: A Trust Assessment Method for In-Vehicular Networks using Vehicle Risk Assessment

Artur Hermann
Ulm University, Institute of Distributed Systems
artur.hermann@uni-ulm.de

Nataša Trkulja
Ulm University, Institute of Distributed Systems
natasa.trkulja@uni-ulm.de

Anderson Ramon Ferraz de Lucena
DENSO AUTOMOTIVE Deutschland GmbH
a.ferraz@eu.denso.com

Alexander Kiening
DENSO AUTOMOTIVE Deutschland GmbH
a.kiening@eu.denso.com

Ana Petrovska
Huawei Technologies
ana.petrovska@huawei.com

Frank Kargl
Ulm University, Institute of Distributed Systems
frank.kargl@uni-ulm.de

Abstract—**Future vehicles will run safety-critical applications that rely on data from entities within and outside the vehicle. Malicious manipulation of this data can lead to safety incidents. In our work, we propose a Trust Assessment Framework (TAF) that allows a component in a vehicle to assess whether it can trust the provided data. Based on a logic framework called Subjective Logic, the TAF determines a trust opinion for all components involved in processing or forwarding a data item. One particular challenge in this approach is the appropriate quantification of trust. To this end, we propose to derive trust opinions for electronic control units (ECUs) in an in-vehicle network based on the security controls implemented in the ECU, such as secure boot. We apply a Threat Analysis and Risk Assessment (TARA) to assess security controls at design time and use run time information to calculate associated trust opinions. The feasibility of the proposed concept is showcased using an in-vehicle application with two different scenarios. Based on the initial results presented in this paper, we see an indication that a trust assessment based on quantifying security controls represents a reasonable approach to provide trust opinions for a TAF.**

I. Introduction

Future vehicles will use data from various sensors within the vehicle, such as a global navigation satellite system (GNSS) sensor or from neighboring vehicles via Vehicle-to-Vehicle communication. An application running in a vehicle and using its position from local sensors and the positions of surrounding vehicles is cooperative adaptive cruise control (CACC). CACC uses this data to decide whether the ego vehicle should accelerate or decelerate. If the input data of the CACC application is compromised, safety-critical incidents can occur [9].

Modern vehicles are complex networks of electronic control units (ECUs), which creates complex information flows. However, having complex information flows inside vehicles and even more complex ones in cooperative systems makes this data vulnerable to tampering by malicious entities. In today's systems, there exist already many mechanisms to detect and protect against compromised components or data items, e.g., an intrusion detection system. These mechanisms often work independently of each other in different components and can not detect a full range of attacks. Furthermore, even when the detection mechanisms detect malicious activities, it is challenging to assess in which degree this impacts the integrity of the components or data items and what appropriate reaction strategies should be. In the worst case, the system or application could be shut down due to the detection of malicious activities, even thought this is not necessary. Modelling the trust dependencies between components involved in the data flow of a data item taking into account all existing detection and protection mechanisms in these components can help to assess the trustworthiness of the data item in a better way than is done today. We therefore suggest to implement a trust assessment framework (TAF) inside vehicles to enable this assessment. This framework will be built on top of Subjective Logic and subjective trust networks. The TAF assesses the trustworthiness of all ECUs in the data flow chain of a data item to decide whether this data item is trustworthy w.r.t. its integrity not being compromised [16].

A core element, but also an open research question, is how to collect and quantify evidence on trustworthiness. One source can be knowledge about the existence of security controls to protect integrity of data, e.g., the integrity protection of data on a vehicle bus with MACsec. Depending on the security controls implemented in the ECU, there is a different risk of the ECU being compromised. Therefore, security controls are a mandatory source to assess the trustworthiness of an ECU. Knowledge about the applied security controls can come from Threat Analysis and Risk Assessments (TARA), which are conducted anyway at design time for an ISO 21434-compliant vehicle design, as well as from additional run time checks. This paper investigates how such data can be leveraged by a TAF to assess trustworthiness of an ECU and, based on that, the trustworthiness of received input data.

Problem: Knowledge about applied security controls in an ECU can be used to assess the trustworthiness of that ECU.

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.23016
www.ndss-symposium.org

For this purpose, a quantification mechanism is necessary that takes evidence about applied security controls into account.

Solution: We use a TARA to assess security controls relevant to an ECU to protect the integrity of a data item. The TARA-based information created at design time and additional validation checks on the security controls conducted at run time are used to assess the trustworthiness of an ECU.

Contribution: We describe an approach to quantify trustworthiness of an ECU based on the applied security controls. Furthermore, we showcase the feasibility of this approach for an in-vehicle application in two different scenarios.

II. Related Work

There are numerous security mechanisms for in-vehicle networks. MACsec [4], for example, is a security protocol providing confidentiality and integrity for in-vehicle networks. TLS provides end-to-end encryption in in-vehicle networks for applications running in two different ECUs [18]. A hardware security module (HSM) integrated into an ECU can manage and store cryptographic keys and execute cryptographic functions without the application having access to the keys [2]. Secure boot can be used for in-vehicle operating systems, which checks the integrity of the software components relevant to the boot process [12]. Control-flow integrity can be used to protect applications. This is a run time operational assurance mechanism that detects operations modified by an attacker [3].

The aforementioned, but also further security controls can be taken into account for trust assessment in in-vehicle networks. There exist already works on trust assessment in the V2X domain. For example, Garlichs et al. [8] propose a trust assessment for vehicle platooning. They create a trust opinion on a vehicle by comparing the actual behavior of the sender with the information received from the sender. Based on the trust of the host vehicle on its predecessor in the platoon, the host vehicle adjusts its safety distance. Van der Heijden et al. [17] and Diezel et al. [6] used Subjective Logic to fuse the output of several misbehavior detectors of a misbehavior detection system. In this way, they were able to integrate an arbitrary number of misbehavior detectors into their system to enhance detection accuracy. Müller et al. [14] also used Subjective Logic to create trust opinions based on the output of misbehavior detectors. Here, consistency checks of received messages are used to detect inconsistent messages. If inconsistent messages are detected, the trust opinions of the corresponding nodes are adjusted.

In addition to the V2X domain, there are further prior efforts in various domains that have utilized logic and formal, mathematical frameworks for the purpose of trust computation. Firoozi et al. [7] propose two novel in-network data processing schemes based on Subjective Logic for trust management using reputation based systems in distributed wireless sensor networks. Their aim was building a technique that can eliminate or reduce the redundant information in the large volumes of sensed data in the present-day wireless sensor networks, which in return reduces resource consumption. Similarly, Renubala and Dhanalakshmi in [15], propose a fuzzy logic-based trust evaluation approach in order to obtain secured routing in wireless networks. The authors argue that the existing trust-aware routing protocols use long-established cryptographic techniques that no longer suffice in tackling serious security problems in the state of the practice. Lastly, Akhuseyinoglu et al. [1] present a trust management framework that automatically computes the trust of “things” as part of IoT services. Their solution uses Multi-Attribute Decision Making and Subjective Logic to take into account the uncertainties in the trust values.

As presented from the previous works, trust assessment based on the Subjective Logic Framework has already proven its potential, especially in conjunction with misbehavior detection and reputation based systems. However, many other possible sources of trust evidence have not yet been explored. Therefore, in this paper, we analyze a further source of trust evidence - the security controls implemented in the system.

III. Background

As our approach derives trust evidence from a TARA for trust assessment, we first introduce some basics.

A. Threat Analysis and Risk Assessment (TARA)

The ISO 21434 standard contains a method to identify and assess threats and risks of electronic systems in road vehicles, called TARA. In Figure 1, we provide an overview of the TARA modules [10]. The item definition (1) includes, among others, the description of the function and the operational environment of the analyzed item. Based on the item definition, assets are identified (2) that could lead to a damage scenario if a cybersecurity property gets compromised. For each asset, threat scenarios are determined (3) that would threaten the cybersecurity properties of the asset. For each threat scenario and corresponding damage scenario, the impact is rated (4). In addition, the steps to realize a threat scenario (5) and the attack feasibility (6) to conduct these steps are identified. For each threat scenario, a risk is determined (7), and a risk value is calculated based on the determined feasibility and impact of the threat scenario. Finally, a decision is made on the treatment of the identified risks (8) [10]. For the significant risks that require treatment, security goals are defined (9), which are concept-level cybersecurity requirements. Based on that, suitable security controls are determined (10). Note that (9) and (10) are not part of the TARA process. The derived security controls are integrated into the item definition. Then, a second iteration of the TARA can be conducted to analyze whether the foreseen security controls reduce the risk levels.

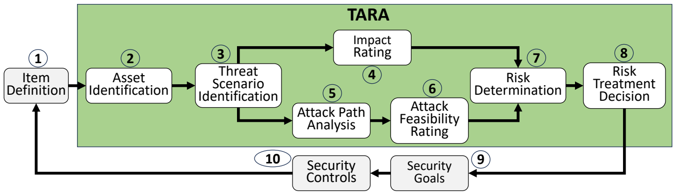

Fig. 1. High-level overview of the TARA process, updated from [13].

B. Subjective Logic

Subjective Logic (SL) is a mathematical logical reasoning framework [11] that can consider the degree of uncertainty of propositions. This uncertainty is represented in SL as part of a *subjective opinion*.

a) Binomial Subjective Opinion: A subjective opinion ω_X^A indicates that the proposition to which the opinion applies is X, and the agent who holds the opinion is A. In this paper, we are using binomial subjective opinions where the propositions can, for example, only be true or false. These opinions are defined as $\omega_x^A = (b_X^A, d_X^A, u_X^A, a_X^A)$, where b_X^A is the belief mass, d_X^A is the disbelief mass, u_X^A is the uncertainty mass, and a_X^A is the base rate. Here, the following equation holds: $b_X^A + d_X^A + u_X^A = 1$. Such a binomial subjective opinion is referred to as a trust opinion in the remainder of the paper.

b) Subjective Trust Networks: The SL framework introduces the concept of *subjective trust networks* (STN). An STN is a directed graph that represents trust relationships from agents via other agents to target entities, where each trust relationship is expressed as a trust opinion [11]. We show a simple STN on the left side of Figure 2. An STN with additional metadata necessary for the trust assessment is referred to as a *Trust Model* in the remainder of the paper [16].

SL knows several operators useful for evaluating an STN. In this paper, however, we restrict ourselves to the use of the *trust discounting* operator [11]. Trust discounting is the process of deriving trust from transitive trust paths. An example of this is shown on the right-hand side of Figure 2, where the trust opinions ω_B^A and ω_X^B are discounted to $\omega_X^{[A;B]}$.

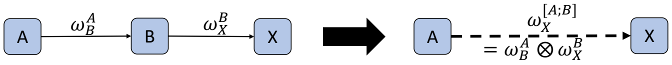

Fig. 2. Subjective trust network (left) and discounted trust opinion (right).

IV. Running Example

This section introduces a running example based on which the trust assessment is illustrated. Figure 3 shows an in-vehicle network. All components in this network are ECUs that can create or process data. The figure also shows the data flow of the position data from the GNSS ECU to the vehicle computer (VC). We assume that a CACC application is running in the VC that wants to assess the trustworthiness of the position data originally provided by the GNSS ECU but forwarded by multiple intermediary entities. Throughout the rest of the paper, we will refer to the position data just as *position*.

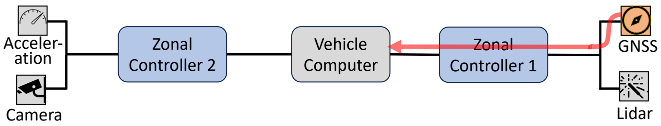

Fig. 3. In-vehicle network with the data-flow of position data.

Based on the in-vehicle network, a Trust Model can be derived. In the Trust Model, all data items, i.e., the position and the ECUs creating and forwarding the position, are represented as nodes. As shown in Figure 4, the Trust Model has three trust relationships in the trust chain (ω_{ZC1}^{VC}, ω_{GNSS}^{ZC1}, and ω_{Pos}^{GNSS}). We assign a trust opinion ω to each trust relationship as part of the Trust Model. In this paper, we focus on how the vehicle computer (VC) can form a trust opinion on the zonal controller 1 (ZC1) by using knowledge about security controls implemented in ZC1. The same approach could be used for the other trust relationships between ECU nodes in the Trust Model. However, for trust relationships involving data items, other sources of evidence are necessary.

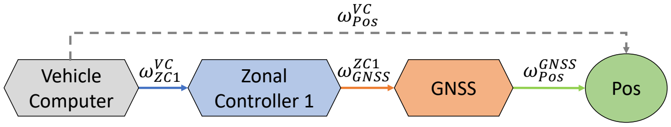

Fig. 4. Trust Model derived from the in-vehicle network.

To assess the trustworthiness from the VC to the ZC1, the VC analyzes design time and run time information about the security controls in ZC1 to calculate the trust opinion ω_{ZC1}^{VC}. This opinion refers to the proposition that the integrity of the position forwarded by ZC1 was not compromised in ZC1. The final goal is to create the opinion ω_{Pos}^{VC} that the VC has on the position even though the VC does not observe the position directly (i.e., it does not have a direct relationship with the position). This is done by discounting the calculated opinions on the three trust relationships in the trust chain ($\omega_{Pos}^{VC} = \omega_{ZC1}^{VC} \otimes \omega_{GNSS}^{ZC1} \otimes \omega_{Pos}^{GNSS}$). ω_{Pos}^{VC} is then compared with a threshold value to determine if the position is trusted.

V. Assessing Trust Based on Security Controls

Depending on the security controls implemented in the ECU, there is a different level of risk of the ECU or the data provided by this ECU being compromised, as security controls can prevent but also detect malicious activities. Therefore, security controls are an essential source of evidence to assess the trustworthiness of an ECU. For this purpose, we describe an approach to quantify trust based on the implemented security controls, which is summarized in Figure 5. In this approach, the system knowledge is used at design time to determine the risk levels in the ECU. Based on these risk levels and the evidence collected at run time about implemented controls, the trust opinion for the ECU is calculated. The single steps of this approach are described in the following.

A. Security Control and Risk Level Determination (Step 1)

In the first step of our approach, a TARA is conducted at design time for the ECU for which the trust opinion is determined, ZC1 in our example. All components within ZC1 that process the position and transmit it to the VC are relevant assets analyzed by the TARA. As we focus here on the integrity of the position, the relevant assets are analyzed regarding the integrity property. For the relevant assets, risks are derived. The ISO 21434 standard proposes risk levels

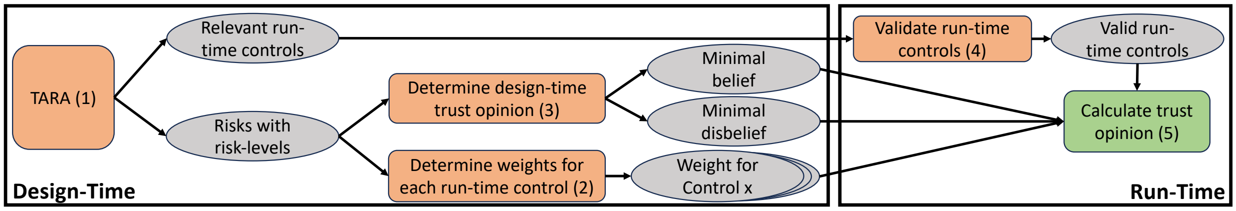
TARA (1)
Relevant run-time controls
Risks with risk-levels
Determine design-time trust opinion (3)
Determine weights for each run-time control (2)
Minimal belief
Minimal disbelief
Weight for Control x
Design-Time
Validate run-time controls (4)
Valid run-time controls
Calculate trust opinion (5)
Run-Time

Fig. 5. Overview of the trust assessment approach based on the output of the TARA.

between one and five [10], while we use levels between zero and four in this paper as this simplifies further calculations.

Based on the identified risks, security controls are determined and integrated into the item definition (see steps (1), (9), and (10) in Figure 1). We distinguish here between controls whose existence is only known from design time because they can not be validated during run time (design time controls) and controls that can be validated during run time (run time controls). For run time controls, evidence is collected from the ECU, which is assessed based on these security controls. This could, for example, be realized with *Direct Anonymous Attestation with Attributes* (DAA-A) [5]. However, the details are out of scope of this paper. We assume that positive or negative evidence can be provided. Positive evidence shows that the control is active and in a valid state, i.e., it has been configured correctly, is working as expected, and has not been compromised. Negative evidence shows that a security control is not active or is not in a valid state.

The integration of security controls and the analysis of risk levels will be done in two steps. In the first step, only design time controls are added to the item definition. These controls are either selected in the TARA process based on the identified risks or are already present in the ECU, as ECUs usually already have some controls implemented in their default setup. Based on all selected design time controls, one TARA iteration is conducted, and a risk level is calculated for each risk.

In the second step, run time controls are analyzed. If a run time control is selected in the TARA process, this control is added to the item definition, and a further TARA iteration is done to determine the risk level when this control is applied. Furthermore, a TARA iteration is conducted for each run time control that is already included in the default setup of the ECU. If these security controls have an impact on the risk levels of the identified risks, these controls are also included in the list of relevant security controls and will be validated during run time. In each TARA iteration, only the design time controls and the currently analyzed run time control are included in the item definition. Thus, each run time control is analyzed in isolation to determine its effect.

Running Example: Table I shows the identified risks (rows) and the corresponding security controls (columns). The first column represents the first step, in which the design time controls are taken into account. The other columns represent the run time controls. These security controls are described in Section II. The values in the table are the corresponding risk levels. The last row shows the sum of all risk levels for the case that the respective control is implemented. We note that no actual TARA was conducted to create the table but that the risks and corresponding risk levels are just examples for illustration purposes.

TABLE I
RISKS AND RISK LEVELS FOR THE TRUST RELATIONSHIP TO ZC1.

Risk	**Original Risk**	**MACsec: C1**	**Secure Boot: C2**	**C.-F. Integrity: C3**	**TLS: C4**	**HSM: C5**
R1 (Compromised Application)	3	2	1	0	2	3
R2 (Compromised OS)	3	2	1	3	2	3
R3 (MitM external comm.)	4	1	4	4	1	4
R4 (Compromise Network Firmware)	2	1	2	2	1	2
R5 (MitM internal comm.)	1	1	1	1	0	1
R6 (Impersonation attack)	4	2	4	4	2	1
R7 (Physical attack on RAM)	1	1	1	1	1	1
Total Risk	18	10	14	15	9	15

B. Weight Calculation (Step 2)

To assess the trustworthiness of an ECU based on security controls, in the second step the importance of the run time controls is assessed. For this purpose, weights are assigned to each control. These weights are values between zero and one and are used to determine how much the trust opinion will be adjusted depending on the run time evidence provided for this control. The weight for each security control is determined in three steps: First, the risk levels of all identified risks are summed up when only the design time controls are applied. Second, the associated risk levels of all identified risks are summed up when the corresponding run time control is applied. Third, a weight is calculated based on the difference between these two values. Depending on how many risks the corresponding run time control affects and how much the risk levels are reduced, the more the control protects the system against identified risks. Thus, the more the risk levels in all risks are reduced, the higher is the weight for the control. Equation 1 shows the calculation of the weights. Here *C* represents the set of all security controls. The variable $sumRisks_{C_x}$ is the sum of all risk levels associated with the implementation of C_x, e.g., $sumRisks_{C_1}$ is the sum of risk levels if security control one is applied. $sumRisks_{\emptyset}$ is the sum of risk levels if no security controls are applied.

$$W_{C_x} = \frac{sumRisks_{\emptyset} - sumRisks_{C_X}}{\sum_{N=1}^{|C|}(sumRisks_{\emptyset} - sumRisks_{C_N})} \quad (1)$$

Running Example: The weight calculation for the security control MACsec and the weights for the other controls are shown below.

$$W_{C_1} = \frac{18-10}{(18-10)+(18-14)+\cdots} = \frac{8}{8+4+3+9+3} = 0.30$$
$$W_{C_2} = 0.15\,,\ W_{C_3} = 0.11\,,\ W_{C_4} = 0.33\,,\ W_{C_5} = 0.11$$

C. Design Time Trust Opinion Calculation (Step 3)

Based on the design time information (DTI), i.e., the determined risk levels and the relevant run time controls, ω_{DTI} is calculated. This opinion is used as a starting point for trust assessment based on run time information and represents the minimal belief and minimal disbelief in the system based on DTI. This trust opinion is necessary because even if all design time and run time controls are applied, there is still a remaining risk and therefore a non-zero disbelief in the system. On the other hand, the risk levels are already low in some systems because the design time controls already reduce the risks. So even if no positive evidence for the run time controls is provided during run time, there could still be a belief in the system. These two aspects are reflected in ω_{DTI}.

a) DTI-based belief: The DTI-based belief represents the minimum belief in the system. For this purpose, a worst-case analysis is conducted by calculating the maximum possible disbelief d_{max}. We assume full knowledge about all controls (none of the run time controls are applied) so that the uncertainty is zero. As belief, disbelief, and uncertainty add up to one, the DTI-based belief can be calculated with Equation 2.

To calculate d_{max}, the risk levels are analyzed when only the design time controls are applied, as this is known from design time, and all run time controls provide negative evidence. Here, a combination of the maximum and the average risk levels of all risks is used. Using only the maximum risk level has the disadvantage that risks with a high risk level have a high impact on d_{max}. This is problematic, as very few high level risks do not represent the security situation of the overall system. Therefore, the highest risk level should only impact d_{max} in a limited way. However, using only the average risk levels is also problematic. If there are many risks with a low risk level, this will result in a low average risk level and a low d_{max} value, even though there are several risks with a high risk level. Therefore, we combine both approaches. Based on the maximum risk level, the first d_{max} value is calculated. For example, when the maximum risk level is 4, this results in $d_{max} = 0.75$, as the d_{max} value is set to increase in 0.25 steps (see Equation 3). Then, a second d_{max} value is calculated, which increases linearly with the average risk level. For example, if the average risk level is 2, the second d_{max} value will be 0.5, as the average risk level is divided by the scaling factor $maxRisk = 4$. From these two values, the maximum value is selected.

b) DTI-based disbelief: The DTI-based disbelief represents the minimum disbelief in the system. For this purpose, a best-case analysis is conducted. Here, the risk levels are analyzed for the case that the design time controls are applied and positive evidence was provided for all run time controls. The approach is the same as for the calculation of the maximum disbelief, with the difference that here the risk levels are taken into account for the case that all run time controls are applied. This approach is represented in Equation 4.

Based on b_{DTI} and d_{DTI}, the DTI-based uncertainty u_{DTI} is calculated as shown in Equation 5.

$$b_{DTI} = 1 - d_{max} \quad (2)$$

$$\begin{aligned} d_{max} = max\{&0.25 \times (maxRisk_{NoControls} - 1), \\ &avgRisk_{NoControls}/maxRisk\} \end{aligned} \quad (3)$$

$$\begin{aligned} d_{DTI} = max\{&0.25 \times (maxRisk_{AllControls} - 1), \\ &avgRisk_{AllControls}/maxRisk\} \end{aligned} \quad (4)$$

$$u_{DTI} = 1 - b_{DTI} - d_{DTI} \quad (5)$$

Running Example: Based on the Equations 2, 3, 4, and 5, ω_{DTI} can be calculated. Using the risk values in Table I results in $d_{max} = 0.75$ and thus in $b_{DTI} = 0.25$. Furthermore, $d_{DTI} = 0.18$ and $u_{DTI} = 0.57$ can be calculated based on the risk values. Based on ω_{DTI} and run time evidence, ω_{ZC1}^{VC} is calculated. If negative evidence is provided for all run time controls, there is still a belief of 0.25, but a disbelief of 0.57 will be added to d_{DTI}. If positive evidence is provided for all controls, the disbelief is still 0.18, while 0.57 belief is added. This approach is described in more detail in the following.

D. Trust Opinion Calculation (Step 4 + 5)

The relevant run time controls, the weights for each control, and ω_{DTI} are determined at design time and stored in the Trust Model so that the TAF can access them at run time.

Based on ω_{DTI} and depending on if positive or negative evidence is received for a control, either the belief or disbelief will be increased, and the uncertainty will be decreased by the same amount. How much the belief or disbelief will be increased depends on the specified weight of the control and u_{DTI} (see Equation 6). u_{DTI} is used in this equation as this is the uncertainty of the ECU, which is reduced with received evidence and thus knowledge about the ECU. For disbelief and uncertainty, the same equation is used as for the belief, as the value for these three attributes is the same. If no evidence is provided for a security control, e.g., because it could not be provided within the time requirements, neither belief nor disbelief will change. This results in a higher uncertainty for the final trust opinion as u_{DTI} has a high uncertainty.

$$\Delta b_{C_x} = \Delta d_{C_x} = \Delta u_{C_x} = W_{C_x} \times u_{DTI} \quad (6)$$

Running Example: We assume that positive evidence was provided for almost all run time controls. Negative evidence was provided for control-flow integrity (C_3) as this control is not used in the corresponding application. Furthermore, no evidence was provided for HSM (C_5), so that we do not know if it is in a valid state. Based on the DTI-based trust opinion and the calculated weights, we can derive the trust opinion w_{ZC1}^{VC}. The amount of belief/disbelief added by each control is calculated with Equation 6 and is $\Delta b_{C_1} = \Delta u_{C_1} = 0.17$, $\Delta b_{C_2} = \Delta u_{C_2} = 0.09$, $\Delta d_{C_3} = \Delta u_{C_3} = 0.06$, and $\Delta b_{C_4} = \Delta u_{C_4} = 0.19$. The calculation of ω_{ZC1}^{VC} is visualized in Figure 6 resulting in the opinion $\omega_{ZC1}^{VC} = (0.7, 0.24, 0.06)$.

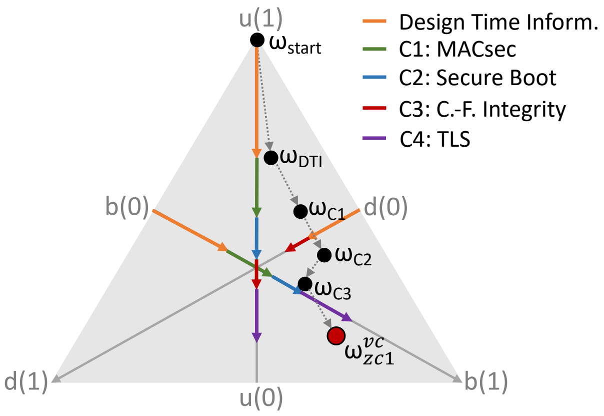

Fig. 6. Calculation of a trust opinion based on implemented security controls.

VI. Preliminary Demonstration

To demonstrate the feasibility of our approach, we show that the TAF derives a correct decision on whether the received position is trustworthy based on the trust opinion ω_{ZC1}^{VC} calculated with our approach. For this purpose, the trust opinion ω_{ZC1}^{VC} is calculated for two scenarios: 1) scenario where positive evidence is provided for most foreseen run time controls in ZC1 and 2) scenario where negative evidence is provided for many run time controls in ZC1. For the other trust relationships in the running example, we assume fixed trust opinions: $\omega_{GNSS}^{ZC1} = (0.9, 0.05, 0.05)$ and $\omega_{Pos}^{GNSS} = (1.0, 0.0, 0.0)$. For all trust opinions, a base rate is assigned, i.e., the prior probability for a proposition in absence of evidence. The base rate can, for example, be derived from past observations and adjusted during run time [11]. We set the base rate to $a = 0.1$ as we argue that the VC has only few past observations on ZC1, e.g., because the CACC application and thus the communication with ZC1 has only recently started.

Based on these three opinions, the TAF running in the VC can calculate via the trust discounting operator the trust opinion it has on the provided position (see ω_{Pos}^{VC} in Figure 4). ω_{Pos}^{VC} is compared with a threshold value to decide if the position is trusted. The calculation of this threshold is out of scope of this work. We set it to $th = 0.6$ for this paper.

a) Trustworthy Scenario: The first scenario and the calculation of ω_{ZC1}^{VC} were already described in Section V-D. The trust opinion from the VC to the position can be calculated based on the three trust opinions in the trust chain.

$$\omega_{Pos}^{VC} = \omega_{ZC1}^{VC} \otimes \omega_{GNSS}^{ZC1} \otimes \omega_{Pos}^{GNSS} = (0.64, 0.0, 0.36)$$

Based on the projected probability provided by SL [11] and the base rate $a = 0.1$, we can calculate a probabilistic value out of ω_{Pos}^{VC}, which is $p_{Pos}^{VC} = b_{Pos}^{VC} + a \times u_{Pos}^{VC} = 0.68$. This value is compared with $th = 0.6$. As p_{Pos}^{VC} is greater than th, the position is considered trustworthy. Since positive evidence has been provided for many foreseen run time controls, many risks are mitigated. Thus, the risk that the position provided by ZC1 is compromised appears to be low. Therefore, the position provided by ZC1 is trusted.

b) Untrustworthy Scenario: In the second scenario, positive evidence was provided for MACsec. Negative evidence was provided for secure boot, control-flow integrity, and HSM. Furthermore, there is no evidence for TLS. Based on the provided evidence, the opinion $\omega_{ZC1}^{VC} = (0.42, 0.39, 0.19)$ is derived. Using trust discounting results in $\omega_{Pos}^{VC} = (0.4, 0.0, 0.6)$ and the projected probability $p_{Pos}^{VC} = 0.46$. As p_{Pos}^{VC} is smaller than th, the position is not considered trustworthy. Since negative evidence has been provided for many foreseen run time controls, many risks are not mitigated. Thus, the risk that the position provided by ZC1 is compromised appears to be high. Therefore, the position provided by ZC1 is not trusted.

VII. Emerging Research Directions

In this paper, we focus on one source of trust evidence that uses design time information provided by a TARA and run time information of security controls to calculate a trust opinion for a trust relationship of a Trust Model. However, in our proposed TAF, further sources of trust evidence could be used to calculate a trust opinion for a trust relationship. This could lead to more accurate trust opinions as more evidence is used for the trust assessment. Such sources of trust evidence can be all security processes and mechanisms, e.g., misbehavior detection systems, intrusion detection systems or spoofing detection systems. For each source of trust evidence, an approach is necessary that interprets and analyzes this evidence and calculates a trust opinion based on it. The trust opinions calculated based on the single sources of trust evidence are then fused together by a fusion operator resulting in the final trust opinion for the corresponding trust relationship. Such fusion operators are provided by the Subjective Logic Framework [11]. An overview of this approach is shown in Figure 7.

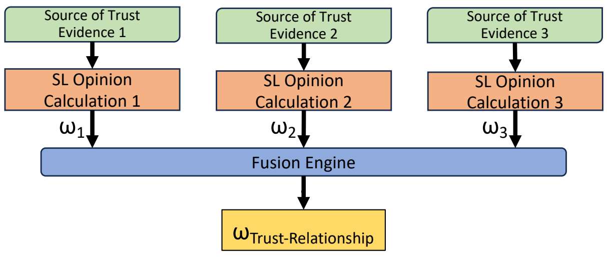

Fig. 7. Calculation of a trust opinion based on several sources of trust evidence.

Although the described sources of trust evidence and our solution for dynamically assessing trust that we propose as part of this paper are mainly tailored for the automotive domain, in theory we aim to build a generalizable TAF that can ideally be extended to use cases and application scenarios from other domains. Examples for further domains are other cyber-physical systems (e.g., robots), IoT, data security (e.g., Usage Control), networking domain (e.g., trusted path routing and 6G), and many more.

VIII. Discussion

Due to the ongoing nature of our presented work, there are open aspects and limitations that will be discussed in this section and elaborated on in future work. As described above, the design time controls are taken into account to calculate ω_{DTI}. Here, it is assumed that the design time controls are implemented and active in the ECU. However, as this can not be validated for these controls during run time, this should be reflected in the uncertainty of the final trust opinion. Thus, an approach is necessary to quantify the uncertainty caused by the lack of run time validation of design time controls.

To determine the weights for the run time controls, we use a TARA to analyze each run time control individually. However, in some situations, security controls when used in combination might have a higher combined impact than their individual contribution. Therefore, a potential solution would be to conduct a TARA not only for each security control individually, but also for combinations of security controls. In this way, weights for combinations of security controls could be tailored to this fact.

Another aspect is that some security controls can detect attacks. Thus, they can provide output during run time that they have detected malicious activities. An example of this is control-flow integrity. This output can also be taken into account to assess trustworthiness. However, a separate approach is probably required for this purpose. Even malicious activities detected by one security control could lead to the case of full disbelief because the output of the control makes it very likely that the system or data item has been compromised.

IX. Conclusion

This paper describes an approach that uses applied security controls as a source of evidence for a Trust Assessment Framework (TAF). With this approach, the trustworthiness of an electronic control unit (ECU) in an in-vehicle network is assessed in the form of a trust opinion. For this purpose, the Threat Analysis and Risk Assessment (TARA) is used at design time to derive the foreseen security controls that should be implemented in the ECU and the impact of these controls on the identified risks. During run time, it is analyzed which of the foreseen run time controls are in a valid state. Based on this design time and run time information, a trust opinion is calculated for the corresponding ECU. The trust opinion of the ECU and the trust opinions of the other entities involved in the data flow of a data item are then used to decide whether the data item is trusted or not.

Based on the initial results presented in this paper, we see an indication that our trust quantification approach based on applied security controls represents a reasonable approach to provide trust opinions for a TAF. Therefore, the described approach and the open questions will be analyzed in future works in the context of in-vehicle networks, but also beyond, e.g., in V2X networks where a vehicle wants to assess the trustworthiness of an external entity, such as another vehicle or a road side unit.

Acknowledgment

This work is partially funded by the HORIZON CONNECT project under EU grant agreement no. 101069688 and the ConnRAD project under grand agreement no. 16KISR036.

References

[1] Akhuseyinoglu, N.B., Karimi, M., Abdelhakim, M., Krishnamurthy, P.: On automated trust computation in iot with multiple attributes and subjective logic. In: 2020 IEEE 45th Conference on Local Computer Networks (LCN). pp. 267–278. IEEE (2020)

[2] Apvrille, L., El Khayari, R., Henniger, O., Roudier, Y., Schweppe, H., Seudié, H., Weyl, B., Wolf, M.: Secure automotive on-board electronics network architecture. In: FISITA 2010 world automotive congress, Budapest, Hungary. vol. 8 (2010)

[3] Burow, N., Carr, S.A., Nash, J., Larsen, P., Franz, M., Brunthaler, S., Payer, M.: Control-flow integrity: Precision, security, and performance. ACM Computing Surveys (CSUR) **50**(1), 1–33 (2017)

[4] Carnevale, B., Falaschi, F., Crocetti, L., Hunjan, H., Bisase, S., Fanucci, L.: An implementation of the 802.1ae mac security standard for in-car networks. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 24–28 (2015)

[5] Chen, L., Urian, R.: Daa-a: Direct anonymous attestation with attributes. In: Conti, M., Schunter, M., Askoxylakis, I. (eds.) Trust and Trustworthy Computing. pp. 228–245. Springer International Publishing (2015)

[6] Dietzel, S., van der Heijden, R., Decke, H., Kargl, F.: A flexible, subjective logic-based framework for misbehavior detection in v2v networks. In: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014. pp. 1–6 (2014)

[7] Firoozi, F., Zadorozhny, V.I., Li, F.Y.: Subjective logic-based in-network data processing for trust management in collocated and distributed wireless sensor networks. IEEE Sensors Journal **18**(15), 6446–6460 (2018)

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

[9] van der Heijden, R., Lukaseder, T., Kargl, F.: Analyzing attacks on cooperative adaptive cruise control (cacc). In: 2017 IEEE Vehicular Networking Conference (VNC). pp. 45–52 (2017)

[10] ISO: Iso/sae 21434:2021. p. 88. pub-ISO (2021), https://www.iso.org/standard/70918.html

[11] Jøsang, A.: Subjective logic. Springer (2016)

[12] Kohnhäuser, F., Püllen, D., Katzenbeisser, S.: Ensuring the safe and secure operation of electronic control units in road vehicles. In: 2019 IEEE Security and Privacy Workshops (SPW). pp. 126–131 (2019)

[13] Luo, F., Jiang, Y., Zhang, Z., Ren, Y., Hou, S.: Threat analysis and risk assessment for connected vehicles: A survey. Security and Communication Networks **2021**, 1–19 (2021)

[14] Müller, J., Meuser, T., Steinmetz, R., Buchholz, M.: A trust management and misbehaviour detection mechanism for multi-agent systems and its application to intelligent transportation systems (05 2019)

[15] Renubala, S., Dhanalakshmi, K.: Trust based secure routing protocol using fuzzy logic in wireless sensor networks. In: 2014 IEEE International Conference on Computational Intelligence and Computing Research. pp. 1–5. IEEE (2014)

[16] Trkulja, N., Hermann, A., Petrovska, A., Kiening, A., de Lucena, A.R.F., Kargl, F.: In-vehicle trust assessment framework. In: 21th escar Europe : The World's Leading Automotive Cyber Security Conference (Hamburg, 15. - 16.11.2023) (2023). https://doi.org/10.13154/294-10384

[17] Van Der Heijden, R.W.: Misbehavior detection in cooperative intelligent transport systems. Ph.D. thesis, Universität Ulm (2018)

[18] Zelle, D., Krauß, C., Strauß, H., Schmidt, K.: On using tls to secure in-vehicle networks. ARES '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3098954.3105824