C-TAR: A Compositional Threat Analysis and Risk Assessment Method for Infrastructure-Based Autonomous Driving

Mohamed Abdelsalam[1,2], Simon Greiner[1], Oum-El-Kheir Aktouf[2], and Annabelle Mercier[2]

[1] Robert Bosch GmbH, Abstatt Robert-Bosch-Allee 1, Germany
{mohamed.abdelsalam,simon.greiner@de.bosch.com}
[2] Grenoble University, 621 Av. Centrale, Grenoble, France
{mohamed.abdelsalam,oum-el-kheir.aktouf,annabelle.mercier}@lcis.grenoble-inp.fr

Abstract. Autonomous Vehicles rely heavily on their sensors' information to navigate correctly. Autonomous driving requires the support of infrastructure-based systems to provide extra sensor information, which cannot be collected by vehicles. We expect that such infrastructure-based systems are typically not provided by the same manufacturer as the vehicle using them. In this paper, we propose a first of its kind, compositional threat analysis and risk assessment method, called C-TAR, and illustrate the method using a simplified example from an autonomous driving context. The proposed method extends a common threat and risk analysis method by statements of dependency on interfacing systems and provides a compatibility check of two systems working together. C-TAR allows the user to identify whether two independently developed systems can interact together securely based on the extended threat and risk analysis.

Keywords: Automotive Security · IoT · Autonomous Vehicles · Smart Infrastructure · Compositionality · Threat Analysis and Risk Assessment · TARA · C-ITS.

1 Introduction

Autonomous Vehicles (AVs) need to be aware of their surroundings to navigate streets safely. Sensing a vehicle's environment can be achieved by sensors inside a vehicle or sensor information provided by an IoT network in the infrastructure. Using infrastructure systems is one way of supporting AVs in particular in cases where the vehicle's sensors are limited, e.g., due to occlusions. The setup of infrastructure sensors can be optimized for a certain area, e.g., an urban intersection, such that the local geography is taken into consideration. We can expect that typically, such infrastructure systems are not provided by the same manufacturer or operator as the vehicles using the infrastructure systems.

The key problem in assessing the security of such IoT configurations of AVs connected to smart infrastructure systems is the large variety of different systems

by different manufacturers which might be present on the roads. Currently available approaches for security analysis require the analysis of the overall system, in our case the combination of vehicle and infrastructure system. It is infeasible to provide this analysis for every combination of infrastructure systems and vehicles which may at some point in time decide to connect to them. The potential configurations an AV may assume over its lifetime are unknown and virtually infinite, meaning that the number of vehicle to infrastructure configurations is countless.

Usually, systems are developed independently from each other, either because one system existed before the other or because they were not developed cooperatively, manufacturers perform threat analysis and risk assessment (TARA) to help identify, assess, prioritize, and mitigate security risks of a given system. If each of these systems is secure in the sense that all risks identified are dealt with appropriately, this does not guarantee that this is also true when two systems are combined into a cooperative traffic system. This is due to the fact that, when two systems are connected, new attack paths and threats may arise which could not be covered by the original security analysis. Individual manufacturers cannot identify such threats as each individual system may not be subject to the respective threat.

In this paper, we propose the first of its kind compositional threat analysis and risk assessment method. We introduce C-TAR, a TARA method that starts with creating a TARA according to ISO 21434 [1] then introduces new elements for a TARA of distributed systems. Using these new TARA elements, C-TAR processes this information and checks the compatibility of the systems. Finally, C-TAR produces a compatibility statement about the overall security of two systems working together. This compatibility statement provides information on whether two systems can securely work together. In case of incompatibility, i.e. there are security threats that arose as a result of connecting the two systems, C-TAR provides the reasons of incompatibility. C-TAR allows independent development of systems, and checking compatibility of systems at runtime. We illustrate C-TAR by applying it on a simplified example for infrastructure-supported autonomous driving.

This paper is structured as follows: Section 2 gives an overview of some preliminaries on threat analysis and risk assessment, followed by related work in Section 3. In Section 4 we describe C-TAR and in Section 5 we provide an example to illustrate our method. Finally we conclude the paper in Section 6.

2 Preliminaries

In this section, we provide an overview of TARA according to the ISO/SAE 21434 standard [1]. The standard defines in Clause 15 the general requirements on an automotive TARA. The purpose of a TARA is to perform a systematic identification of threats which a system is exposed to and the risks associated with these threats. The result of a TARA then serves as the basis for decisions on how to deal with the identified risks.

After defining the system and describing its interfaces and functions, the *assets* of the system are identified. An *asset* can be anything worth protecting in the system, and typically includes information, data, functionality and other elements. For each asset, one or more *security properties* which have to be protected are identified. Typical security properties are confidentiality, integrity, and availability of an asset. A *threat* is defined as the non-fulfillment of a security property.

For each threat, a *damage scenario* is provided, which describes the consequences of what happens when a threat is realised. For each damage scenario, an *impact* rating is provided which quantifies the consequences of a damage scenario. This impact rating serves as the cost-part of the risk analysis for a particular threat.

In order to estimate the probability of a threat, i.e. how likely it is that a threat is realised, the standard follows an indirect approach. Instead of directly providing a probability, which for most threats is hard to quantify, the standard requires a rating of the effort an attacker has to invest in order to realise the threat, the so-called *attack feasibility rating* (AFR).

To provide the AFR for a threat, first the attack paths have to be identified, which an attacker could perform in order to realise a threat. An *attack path* is a sequence of actions which an attacker performs, which together lead to the realization of the related threat. After identifying the attack path, for each attack path an AFR has to be provided. Using the rating of an attack path, together with the other elements of the TARA, the overall risk of the system under evaluation can be calculated. The standard offers several methods to determine the AFR, while we limit the presentation here to the attack potential method. The attack potential method requires to rate each attack path with the five attributes of *Elapsed Time*, i.e., the time an attacker requires to perform the attack path, the amount of *expertise* required by the attacker, the level of *knowledge about the product*, the *window of opportunity* required to perform the attack, and finally the type of *equipment* the attacker has to have available.

The standard leaves open how the different categories are translated into an AFR, however, it provides a possible realization as an example in the informative Appendix. Our presentation here is based on that example. Each rated category of an attack path is translated into an attack potential value as shown in table 1. The attack potential values of an attack path are then added and the resulting number is translated according to another table in the AFR values *High*, *Medium*, *Low*, and *Very low*. Finally, the AFR together with the impact of a threat can then be translated into a qualitative risk value.

Table 1. ISO 21434 Attribute Ratings Table [1]

Elapsed time		Specialist expertise		Knowledge of the item or component		Window of opportunity		Equipment	
Enumerate	**Value**	**Enumerate**	**Value**	**Enumerate**	**Value**	**Enumerate**	**Value**	**Enumerate**	**Value**
≤1 day	0	Layman	0	Public	0	Unlimited	0	Standard	0
≤1 week	1	Proficient	3	Restricted	3	Easy	1	Specialized	4
≤1 month	4	Expert	6	Confidential	7	Moderate	4	Bespoke	7
≤6 months	17	Multiple experts	8	Strictly confidential	11	Difficult/ none	10	Multiple bespoke	9
>6 months	19								

3 Related Work

In this section we present and discuss some of the related work that we classify into four main classes: threat modeling, risk assessment, digital dependability identities and automotive security surveys.

3.1 Threat Modeling

According to the work of Casola et al. [5], IoT systems are characterized by the high heterogeneity of involved systems. Moreover, there is a lack of a comprehensive threat model for IoT systems. This makes performing effective security assessment of actual IoT deployments very difficult.

Casola et al. [5] and Rak et al. [17] propose methods to automate risk analysis for IoT systems to identify threats and their related countermeasures. The authors in [5] use the information stored in a security knowledge base which maps threats to assets to build a threat model. While the authors in [17] relies on an open catalogue, for gathering information about threats and vulnerabilities of the IoT system under analysis. The identified threats are then associated with a risk level and mapped to a set of suitable countermeasures. The authors of [17] applied their method on a case study where they were able to automatically build a custom threat model associated with the system where they reported the asset to protect, the associated threat and the security controls.

In the work of Kim et al. [12], they discuss the use of threat modeling method (TMM) to investigate the potential threats to AVs. Also, in their work of [13], they demonstrate how the threat modeling process used in the computer industry, can be adapted and applied in the automotive domain.

The discussed methods focus on performing risk analysis for single systems which results in identifying individual threats specific to each system. While in our work, we focus on threats that arise as a result of two independently developed subsystems working together and forming an overall system.

3.2 Risk Assessment

Automotive development is a highly distributed process with many organizations involved. Full sharing of information is neither desirable nor possible. However, TARA requires a holistic view to cover all potential attack vectors. To address such an issue, Kiening et al. [11] propose a method to allow organizations to perform TARA analysis according to ISO 21434 in a collaborative and joint way while performing partial risk assessments within their scope. They propose the use of *Cybersecurity Interface Agreement* for a TARA to enable sharing appropriate information among involved organizations. The proposed interface agreement is a contractual agreement that requires developing the systems at the same time. In comparison to our work, we present a TARA method to determine the compatibility of an overall system comprising of two subsystems, while not necessarily having the subsystems developed at the same time, but at different points in time.

Eichler et al. [7] propose a method for risk assessment that targets heterogeneous and complex environments. Similarly in our approach, we are targeting heterogeneous systems to perform threat analysis and risk assessment. However, the focus of our approach is on checking the compatibility of two heterogeneous systems working together. On the other hand, their work focus is on flexibility and scalable effort for risk assessment but not on heterogeneous subsystems forming an overall system.

The development of CPS requires interdisciplinary cooperation between different stakeholders to avoid unidentified security threats. Japs et al. [10] present the SAVE method that enables early identification of safety relevant security threats. SAVE supports stakeholders identify security hazards by creating a SysML system model. The main difference to our work is that they use model-based safety engineering (MBSE) to identify security threats. Moreover, their method is designed to be applied in workshops with an interdisciplinary team of stakeholders.

Standards such as ISO/SAE 21434 Road vehicles - Cybersecurity engineering [1] and ETSI [6], focus on engineering secure functions at the vehicle level or analysis of the threats and risks of an Information and Communications Technology (ICT) system. However, automotive engineering projects are highly distributed among many stakeholders. ISO/SAE 21434 introduces the concept of *Cybersecurity Interface Agreement* to address distributed engineering in automotive industry. However, the standard does not provide any guidance on how to perform activities such as TARA in a collaborative, joint way. Moreover, using cybersecurity interface agreement is not suitable to solve our problem as it is tied to developing the subsystems simultaneously. Looking at both standards, there is no clear guidance on how to address risk analysis of heterogenous systems connected together.

3.3 Digital Dependability Identities

The configurations Cyber-Physical Systems (CPS) may assume over its lifetime are unknown and potentially infinite which makes it difficult to assess the de-

pendability of CPS. Hence, the authors of [18], [4], and [2], proposed and worked with the concept of digital dependability identity (DDI) to work as medium for synthesis of heterogeneous dependability information collected from different systems.

Adler et al. [4] and Armengaud et al. [2] use DDIs to check whether autonomous vehicles that come together at runtime can cooperate dependably. DDI is used to monitor the runtime cooperation between the systems and adapt it so that it will remain dependable. Moreover, the DDI concept was applied to truck platooning use case where they make it possible to check which vehicles are permitted to form a platoon.

While their interest in heterogenous systems is a shared interest with our work, the main difference between our research and theirs, is their focus on the dependability property of safety while our focus is on security property. Another difference is their focus on runtime operation.

3.4 Automotive Security Surveys

Luo et al. [16] have conducted a survey on TARA methods in the automotive field such as STRIDE, EVITA, OCTAVE, and BRA. In the survey of Lamssaggad et al. [14], the authors give a short background on the main security issues that hinder Intelligent Transportation Systems (ITS) and they provide a comprehensive analysis of existing security solutions in the literature. Another survey conducted by Lu et al. [15] provides a comprehensive security analysis for vehicular networks. Similarly, Huang et al. [9] provide an in-depth review of the state-of-the-art solutions concerning security and privacy for V2X communications. Alnasser et al. [3] analyze the threats for V2X and some of the available security solutions. Hammi et al. [8] conducted an extensive survey on the different Public Key Infrastructure (PKI) architectures used in C-ITS environments.

Even though the aforementioned surveys discuss different aspects of automotive security, such as TARA methods, risks, threat assessment and security countermeasures, there are no proposed methods for a TARA method of heterogeneous subsystems forming an overall system. Plenty of the TARA methods proposed used in the automotive industry target only a single system for threat analysis and risk assessment. Hence, the main contribution of this paper is providing a TARA method which solves this problem through analyzing the threats and risks of two independent subsystems forming an overall system.

3.5 Insights from Literature Review

In this section we showcased the current state of the art and discussed the main differences between the proposed methods in the literature and the work presented in this paper. Automotive security is discussed at length in the literature from different perspectives and our work is aimed at the overall security of heterogenous subsystems. To the best of our knowledge, the literature does not have a lot of similar works to ours.

4 C-TAR Method Description

This section elaborates and discusses the developed C-TAR method and presents our contribution of the paper. An overview of the method description is shown in Fig.1. The method consists of three phases, each of which is discussed in detail in the following subsections. The first phase of C-TAR starts from a TARA created according to ISO 21434 and identifying attack paths and extracting the needed information from such attack paths. Followed by that, is the processing phase and checking the compatibility conditions. Finally, the output phase which presents the compatibility statement stating whether two subsystems are secure to work together as an overall system or not.

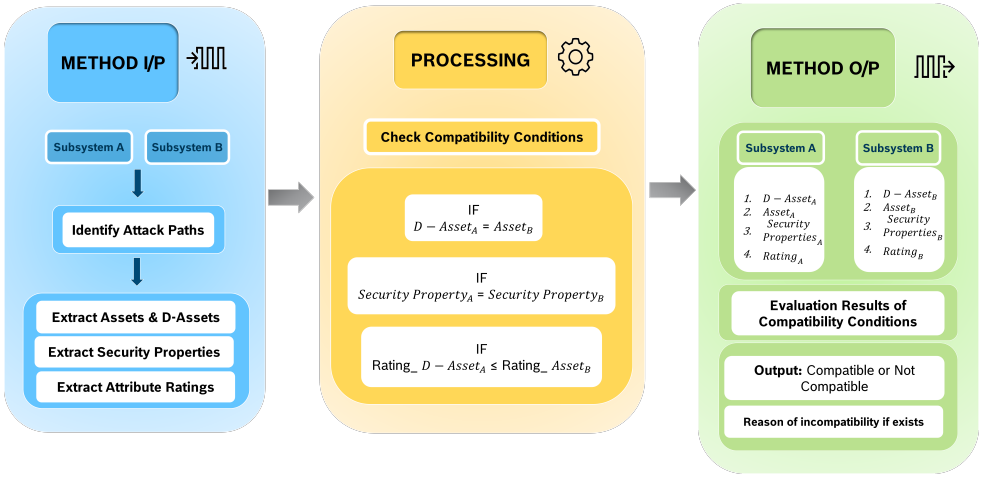

Fig. 1. C-TAR Method Overview

4.1 First Phase - C-TAR Input

When a manufacturer makes a TARA according to the common process described by ISO/SAE 21434 standard, the first phase of C-TAR is to identify the different attack paths for each subsystem under evaluation, in addition to the associated assets, assets security properties and attack path ratings. For every combination of an asset and its security property, the relevant attack paths are identified and given a rating, according to ISO 21434.

While conducting TARA analysis of a subsystem under evaluation, if the TARA of the subsystem under evaluation has a critical connection to another subsystem or dependent on another subsystem, knowledge about the subsystem under evaluation is not sufficient to rate all identified relevant attack paths while creating the TARA. The ISO 21434 standard does not define how to deal with

the missing information in a TARA that is dependent on another subsystem's information.

Hence, we present *dependent attack path*. A *dependent attack path* allows to model parts of a subsystem under evaluation in a TARA, which are unknown or dependent on an external subsystem. To do this, an attack path, which is dependent on an external subsystem, is separated into two parts: A *partial attack path* and a *dependent attack path*. The *partial attack path* models the part of the attack path, which can be determined with knowledge about the subsystem under evaluation alone. This *partial attack path* is rated using the five attributes of AFR mentioned earlier in section 2.

The second part is the *dependent attack path*, which depends on knowledge about the external subsystem. The dependent attack path acts as a placeholder for the part of the attack path which is dependent on the external subsystem information. The *dependent attack path* is given a rating in the subsystem under evaluation as an assumption, this rating becomes a requirement to the external subsystem to satisfy, to guarantee compatibility between the two subsystems. In addition to that, we extract from the *dependent attack path* the *D-asset* and its associated security property.

To reflect dependencies between a subsystem under evaluation and an external subsystem, we define the notion of *dependent attack path*. A *dependent attack path* $(D - asset_x, property_x, (r_{1x}, ..., r_{nx}))$ is a tuple consisting of:

- **Dependent Asset** $(D - asset_x)$: Is something to protect, typically an entity of information or data, this asset typically outside the control of the subsystem under evaluation.
- **Security Property** $(property_x)$: The security property of the dependent asset which has to be broken in order to realise an attack path,
- **Rating** $(r_{1x}, ..., r_{nx})$: The attack path rating given in the TARA for the estimated effort an attacker needs to break the security property,

where x is a notation for the subsystem under evaluation. While a partial attack path, is a set of actions that realizes a threat in conjunction with a dependent attack path.

4.2 Second Phase - Compatibility Conditions

Using the extracted information from the first phase of C-TAR, the second phase of the method is to process this input information and perform a compatibility check for the two subsystems. In order to identify if two subsystems, a subsystem under evaluation and an external subsystem, can be combined in a way that result in an overall system which has at most the risk of the two subsystems, we have to check whether the assumptions, modeled in a dependent attack path, of each of the subsystems, can be guaranteed. By combining the rating of the partial attack path and the dependent attack path, an overall rating of the attack path can be calculated. Using the rating of the attack path, together with the other elements of the TARA, the overall risk of the system under evaluation

can be calculated. The risk is correct if all attack paths related to the dependent attack path are at least as hard as the dependent attack path rating assumption.

In the following, we describe how this compatibility check can be performed.

Two subsystems A and B are compatible, if for every dependent attack path $(D - asset_A, property_A, (r_{1A}, ..., r_{nA}))$ in the TARA of subsystem A:

- **Assets Condition:** There exists an asset $Asset_B$ in the TARA of subsystem B where $D - asset_A = Asset_B$, and
- **Security Property Condition:** There exists a security property $property_B$ for $asset_B$ in the TARA of B where $property_A = property_B$, and
- **Rating Condition:** For every attack path for $asset_B$ and $property_B$ with rating $(r_1B, ..., r_nB)$ it holds that $r_{iA} \leq r_{iB}$ for $1 \leq i \leq n$, and
- Vice versa with A and B exchanged in the conditions above.

Basically, the conditions above formalize that for two subsystems under evaluation, every attack on the D-asset assumed in one subsystem, is identified to require more effort in the TARA of the other subsystem, than assumed in the TARA of the first subsystem. As a result, the risk identified in the TARA of each subsystem reflects a risk which is at most as high as for the subsystem in combination with the other subsystem.

4.3 Third Phase - Compatibility Statement

Applying the aforementioned conditions enables C-TAR, in its third phase, to produce a compatibility statement about the risk assessment of the overall system. The compatibility statement presents an overview of the subsystems under evaluation, the identified and extracted information of a TARA, in addition to the verdict about the compatibility of the two subsystems. The statement starts with presenting one D-asset or several D-assets extracted from the corresponding dependent attack path, in addition to the asset or assets of each subsystem. Moreover, the associated security property of the D-asset and asset are included in addition to the attack paths ratings. Secondly, it presents each of the compatibility conditions and whether they were satisfied or not. Finally, it outputs a statement declaring whether the two subsystems are compatible or incompatible, i.e. whether they form an overall secure system or not. In case the subsystems are incompatible, the reasons of incompatibility are stated in the output.

5 Example

In this section we illustrate C-TAR by applying it to an exemplary development scenario. The given example consists of an AV system and a smart traffic light system (TL) communicating together. Both the AV and the TL are two heterogeneous systems that form an overall connected system. Whereafter, we would refer to the AV and the TL as subsystems while the overall aggregate of both of them as the system, (see Fig.2). The given example is a simplification of a real use case TARA.

Operational Scenario: In the given example, the AV communicates with the TL exchanging information to help it navigate the road safely and to be aware of its surroundings. To initiate communication between the two subsystems, the AV sends a ***REGISTER*** message to register with the TL. While the TL sends a ***SPATEM*** message containing the TL status to the AV, which is either ***RED*** or ***GREEN***. If it is ***RED***, the AV brakes and if it is ***GREEN***, the AV keeps moving. If the AV is within the range of the TL and it does not receive any signal from the TL it switches into safe mode and degrades its speed.

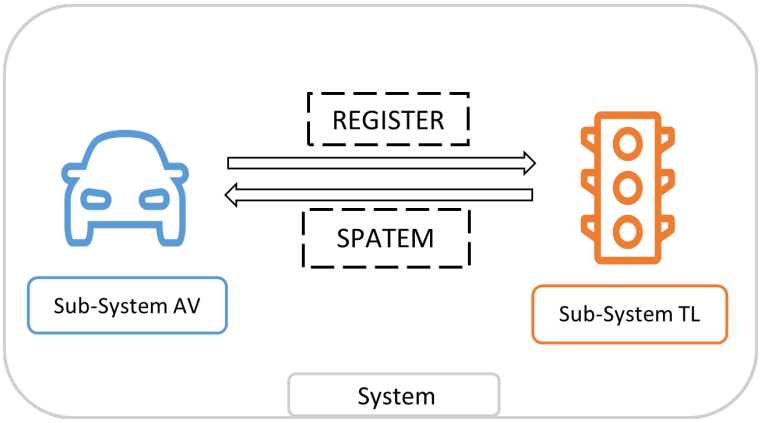

Fig. 2. Given Example Diagram

5.1 First Phase - C-TAR Input

The first phase of the method is creating a TARA according to ISO 21434 with the additional step of identifying attack paths, dependent assets, security properties and attack path rating to use in the second phase of C-TAR for processing. In this example, we have two TARAs for the respective subsystems, TARA_AV and TARA_TL.

TARA_AV

TARA_AV elements:

- **Asset:** Vehicle ECU
- **Security Property:** Integrity
- **Threat:** Manipulate Vehicle ECU

AV Attack Paths: Typically, a TARA of a subsystem, has several attack paths that comprise this TARA. Such attack paths are depicted in Fig.3. There are different types of attack paths shown in the figure. The first type we have is the *dependent attack path*, it contains one action in this example, which is *Steal Certificate*. The second type is the *partial attack path*, it consists of action *Sign Message* and the action *Send message*. In order for an attacker to manipulate the vehicle behavior the following actions by two different attack paths are required. First, in the dependent attack path, in the action *Steal Certificate* an attacker steals the certificate from the traffic light. Second, in the partial attack path, in the first action *Sign Message* an attacker signs a message with the stolen certificate and the wrong traffic light status. The second action in the partial attack path is *Send message* where an attacker sends the signed message via ITS G5 to the vehicle. Since there is a dependent attack path, the next step of C-TAR is the identification of the D-asset, its associated security property and rating of the dependent attack path.

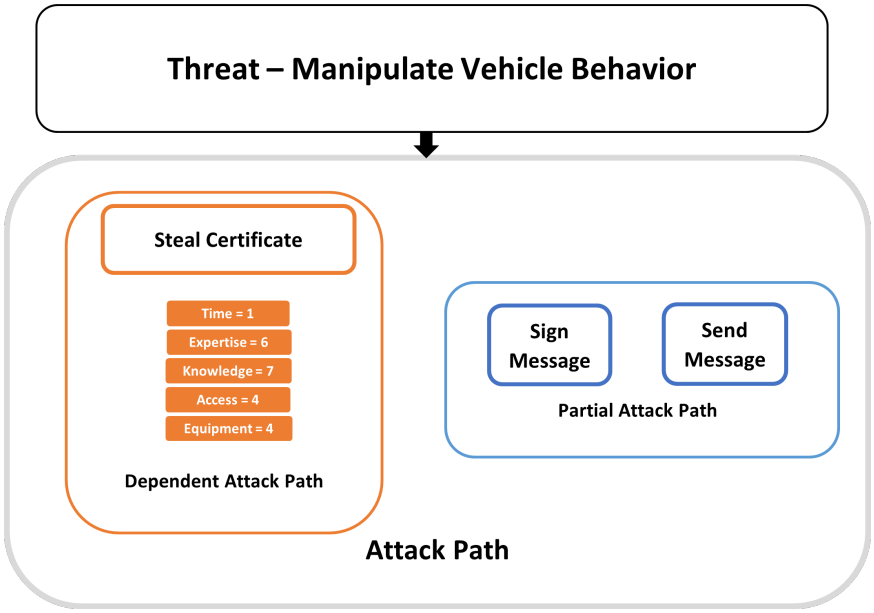

Fig. 3. TARA_AV Attack Paths and Threat

D-Asset: The action *Steal Certificate* is dependent on the TL subsystem. Therefore, the D-asset is identified as *TL Certificate*. Because from the perspective of the AV, the *TL Certificate* is an important asset that is worth protection as it could realise a threat of the AV TARA.

D-Asset Security Property: From the perspective of the AV, the certificate information need to be inaccessable by unauthorized parties, therefore, the security property of this D-asset would be confidentiality.

Rating: Referring to Fig.3, the dependent attack path is given a rating based on assumptions by the AV subsystem about the TL subsystem. As previously mentioned in the method description, the ratings given are based on the ISO 21434 standard. These assumptions are then treated as requirements made by the AV subsystem that need to be satisfied by the TL subsystem. For simplicity reasons we refer to the five attributes of ISO 21434 for rating an attack path as following: "Elapsed time" as time, "Special expertise" as expertise, "Knowledge of the item or component" as knowledge, "Window of opportunity" as access and "Equipment" stays as equipment. The corresponding values for each of the attributes are shown in Fig.3. We estimate less than a week needed by an attacker to perform the attack, this translates to a value of 1 for time. An attacker needs to be an expert which translates the expertise value into a value of 6. The knowledge required is considered to be confidential which translates to a value of 7. Regarding access, it is rated as moderate which translates to a value of 4. Finally, an attacker needs specialized equipment which is rated as 4. Note that these rating values were derived according to ISO 21434 AFR, see table 1.

For the second phase of C-TAR the D-asset and its associated security property, in addition to the dependent attack path rating, are used, see table 2.

Note the difference between the *dependent attack path* and the *partial attack path*. In the *dependent attack path*, there is a dependency on information from the TL subsystem, while for the *partial attack path*, there is no information needed from the TL subsystem.

TARA_TL

TARA_TL elements:

- **Asset:** *TL Certificate*
- **Security Property:** Confidentiality
- **Threats:** Extraction of TL Certificate

TL Attack Paths: The TL TARA has only one attack path in the given example, (see Fig.4). The given attack path consists of the three actions given in the figure. In order for an attacker to extract the TL certificate the following actions are required. First, *Connect to Plug*: in which an attacker connects to the maintenance plug of the traffic light controller. Second, *Brute Force Controller*: an attacker brute forces the traffic light controller maintenance protocol. Third, *Read Certificate*: an attacker reads the certificate from the traffic light flash memory using the maintenance protocol. Since there is no dependent attack path, C-TAR does not identify in the TL TARA a D-asset nor the associated security property.

Rating : The rating values for the attack path according to ISO 21434 is shown in Fig. 4. We estimate the attacker to need less than one month in time which translates into a value of 4. For expertise, an attacker needs to be an expert which translates into a value of 6. The knowledge required by the attacker is considered to be restricted which translates to a value of 3. Regarding access, it is considered to be rated as easy, this translates to a value of 1. Finally, an attacker needs bespoke equipment which is rated as 7. Note that these rating values were derived according to ISO 21434 AFR, see table 1.

For the second phase of C-TAR the asset and security property declared in the TL TARA, in addition to the attack path rating, are used, see table 2.

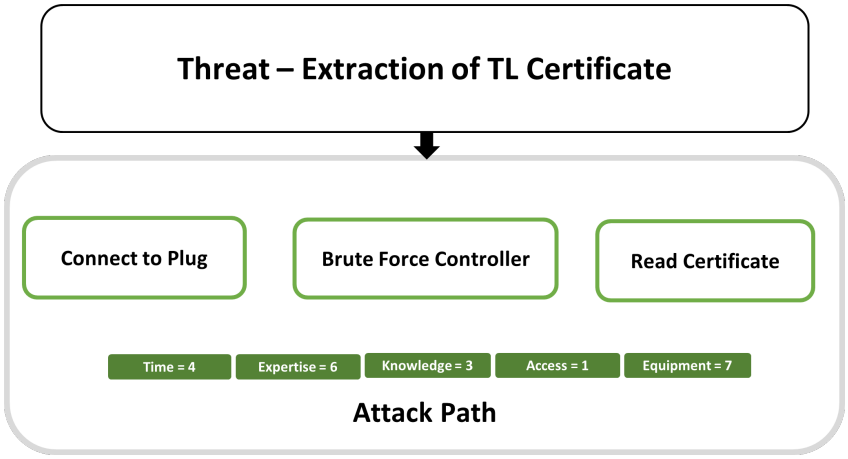

Fig. 4. TARA_TL Attack Path and Threat

Table 2. Generated Data of First Phase Used for the Second Phase of C-TAR

Asset Type	**AV D-asset**	**TL Asset**
Asset Name	*TL Certificate*	*TL Certificate*
Security Property	Confidentiality	Confidentiality
Attack Path Rating	**dependent**	**attack path**
Time	1	4
Expertise	6	6
Knowledge	7	3
Access	4	1
Equipment	4	7

5.2 Second Phase - Compatibility Conditions

In the second phase of C-TAR, we start processing the input and checking the conditions of compatibility, mentioned in section 4.2, on the given input. There are three required conditions of compatibility:

1. **Assets Condition**: $D - asset_{AV} = Asset_{TL}$
2. **Security Property Condition**: $SecurityProperty_{AV} = SecurityProperty_{TL}$
3. **Rating Condition**: $Rating_{AV} \leq Rating_{TL}$

Assets Condition The first condition checks if the D-asset "Steal Certificate from traffic light" of the AV subsystem is considered as an Asset in the TL subsystem TARA. In the given example, the AV's D-asset exists as an asset in TL subsystem, therefore the first condition is satisfied.

Security Property Condition The second condition checks whether the D-asset and asset share the same security properties. Since the D-asset of the AV and the asset of the TL share the same security property of confidentiality, the condition is satisfied.

Rating Condition Finally the third condition is to check if the rating of each one of the five attributes in AV subsystem is less than the rating of its corresponding attribute of the TL subsystem. Note that the rating rates the attack path, not the D-asset or asset. C-TAR compares the attributes ratings of the D-asset to those of the asset. The higher the value of the attribute rating the harder it is for an attacker to realise such attack path and the more secure this path is. The lower the value, the easier it is to perform an attack. Referring to Table 2, the attributes of time, expertise and equipment satisfy the third condition. Contrary to that, the rating of the attributes of knowledge and access in the AV subsystem dependent attack path are greater than that of the TL subsystem attack path. Hence, we conclude that the third condition is not satisfied.

Similar to attack potential or maximum likelihood methods to give a rating in a TARA based on personal evaluation, the dependent attack path is given a rating which might be different when given a rating by a different person.

5.3 Third Phase - Compatibility Statement

In the third phase of C-TAR, the method produces a compatibility statement as an output to the method, (see Fig.5). The compatibility statement of the given example starts by presenting the D-asset of the AV subsystem and the asset of the TL subsystem which is *TL Certificate* for both subsystems. Moreover, it presents the associated ratings of {1, 6, 7, 4, 4} for the AV subsystem and {4, 6, 3, 1, 7} for the TL subsystem, in addition to the security property of confidentiality for the AV D-asset and the TL asset. Following that, the statement displays all needed conditions for compatibility and highlights the satisfied ones in green and

the unsatisfied ones in red. Finally, it provides the verdict regarding the security compatibility of the two subsystems. The output declares the two subsystems incompatible due to not satisfying all of the necessary conditions. The reasons for incompatibility were the knowledge and access rating. In both cases, the AV attribute rating was less than the attribute rating of the TL subsystem, hence not satisfying the third condition.

In conclusion, we can summarize the process as follows: We create a TARA according to ISO 21434, C-TAR checks for attack paths. Then it extracts from the different attack paths; (1) D-asset, (2) associated security property of the different asset types and (3) rating of the attack paths. Afterwards, C-TAR checks conditions of compatibility, if satisfied or not, and finally produces a compatibility statement. For the given example the AV subsystem did not have its requirements satisfied by the TL subsystem guarantees, therefore the two subsystems would not be secure to work together due to not satisfying all conditions.

System Compatibility Statement:

Sub-System 1: AV

D-asset$_{AV}$: *TL Certificate*,
Security Property$_{AV}$: Confidentiality,
Rating$_{AV}$ = {1, 6, 7, 4, 4}

Sub-System 2: TL

Asset$_{TL}$: *TL Certificate*,
Security Property$_{TL}$: Confidentiality,
Rating$_{TL}$ = {4, 6, 3, 1, 7}

Compatibility Conditions:

Condition 1 (Assets Condition):
D-asset$_{AV}$ {TL Certificate} = D-asset$_{TL}$ {TL Certificate}
Condition 2 (Security Property Condition):
Security Property$_{AV}$ {Confidentiality} = Security Property$_{TL}$ {Confidentiality}

Condition 3 (Rating Condition):
Rating_Time$_{AV}$ {1} ≤ Rating_Time$_{TL}$ {4},
Rating_Expertise$_{AV}$ {6} ≤ Rating_Expertise$_{TL}$ {6},
Rating_Knowledge$_{AV}$ {7} ≤ Rating_Knowledge$_{TL}$ {3},
Rating_Access$_{AV}$ {4} ≤ Rating_Access$_{TL}$ {1},
Rating_Equipment$_{AV}$ {4} ≤ Rating_Equipment$_{TL}$ {7}.

Output:

Sub-Systems AV & TL are not compatible.

Reason(s) for incompatibility:

Rating_Knowledge$_{AV}$ > Rating_Knowledge$_{TL,}$

Fig. 5. Compatibility Statement Output

6 Conclusion

In this paper we presented C-TAR, a first of its kind method to help evaluate the threats and risks of an overall system, consisting of two connected heterogeneous subsystems. C-TAR is particularly useful in the automotive IoT domain where there is a network of connected vehicles and infrastructure-based systems communicating and sharing information.

The first step in our approach is to create a TARA according to ISO 21434, then identifies the TARA additional information of dependent attack paths with their attack potential rating and the dependent assets with the security property of these assets to be protected. Based on the dependent assets and attack paths, the compatibility of the two systems can be checked automatically based on well-defined conditions.

C-TAR allows to evaluate the maximum risks and threats for an overall system without actually conducting a TARA for the overall system. Instead, it is sufficient to analyze the constituent subsystems of the overall system separately. Thus, C-TAR is a compositional TARA method. Especially, if two subsystems are developed independently from each other, e.g. as expected for cooperative intelligent traffic systems, applying C-TAR to both subsystems allows to deduce whether the two subsystems are compatible. In case of incompatibility, the reasons provided by C-TAR helps to address the missed threats in the subsystem and provides an explanation why the two subsystems interacting could lead to an unacceptable risk.

In future work, we will apply C-TAR to industrial use cases, to further assess its suitability for larger applications and evaluate its performance. Furthermore, we would compare C-TAR to other methods in the literature in analyzing a specific example.

Acknowledgements The authors would like to thank all partners within the Hi-Drive project for their cooperation and valuable contribution. [This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006664. The sole responsibility of this publication lies with the authors. Neither the European Commission nor CINEA - in its capacity of Granting Authority - can be made responsible for any use that may be made of the information this document contains.]

References

1. ISO/SAE 21434:2021 (Aug 2021), https://www.iso.org/standard/70918.html
2. Adler, R., Reich, J., Kaypmaz, C.: Dependable autonomous commercial vehicles. ATZheavy duty worldwide **14**, 50–54 (2021)
3. Alnasser, A., Sun, H., Jiang, J.: Cyber security challenges and solutions for v2x communications: A survey. Computer Networks **151**, 52–67 (2019)
4. Armengaud, E., Schneider, D., Reich, J., Sorokos, I., Papadopoulos, Y., Zeller, M., Regan, G., Macher, G., Veledar, O., Thalmann, S., et al.: Ddi: A novel technology and innovation model for dependable, collaborative and autonomous systems. In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1626–1631. IEEE (2021)
5. Casola, V., De Benedictis, A., Rak, M., Villano, U.: Toward the automation of threat modeling and risk assessment in iot systems. Internet of Things **7**, 100056 (2019). https://doi.org/10.1016/j.iot.2019.100056
6. CYBER, ETSI: Methods and protocols; part 1: Method and pro forma for threat, vulnerability. Risk Analysis (TVRA). Technical Specification TS **102**, 165–1
7. Eichler, J., Angermeier, D.: Modular risk assessment for the development of secure automotive systems. In: Proceedings of the 31st VDI/VW joint conference Automotive Security, Wolfsburg, Germany. pp. 21–22 (2015)
8. Hammi, B., Monteuuis, J.P., Petit, J.: Pkis in c-its: Security functions, architectures and projects: A survey. Vehicular Communications **38**, 100531 (2022)
9. Huang, J., Fang, D., Qian, Y., Hu, R.Q.: Recent advances and challenges in security and privacy for v2x communications. IEEE Open Journal of Vehicular Technology **1**, 244–266 (2020)
10. Japs, S., Anacker, H., Dumitrescu, R.: Save: Security & safety by model-based systems engineering on the example of automotive industry. procedia CIRP **100**, 187–192 (2021)
11. Kiening, A., Angermeier, D.: Trade-threat and risk assessment for automotive distributed engineering (2021)
12. Kim, S., Shrestha, R.: Chapter 3 Security and Privacy in Intelligent Autonomous Vehicles. Springer (2021)
13. Kim, S., Shrestha, R.: Chapter 5 AUTOSAR Embedded Security in Vehicles. Springer (2021)
14. Lamssaggad, A., Benamar, N., Hafid, A.S., Msahli, M.: A survey on the current security landscape of intelligent transportation systems. IEEE Access **9**, 9180–9208 (2021)
15. Lu, Z., Qu, G., Liu, Z.: A survey on recent advances in vehicular network security, trust, and privacy. IEEE Transactions on Intelligent Transportation Systems **20**(2), 760–776 (2018)
16. 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)
17. Rak, M., Casola, V., De Benedictis, A., Villano, U.: Automated risk analysis for iot systems. In: Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 13th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2018). pp. 265–275. Springer (2019)
18. Schneider, D., Trapp, M., Papadopoulos, Y., Armengaud, E., Zeller, M., Höfig, K.: Wap: digital dependability identities. In: 2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE). pp. 324–329. IEEE (2015)