SARA: Security Automotive Risk Analysis Method

Jean-Philippe Monteuuis
PSA Group / Telecom ParisTech
Velizy, France
jeanphilippe.monteuuis@mpsa.com

Aymen Boudguiga
CEA LIST
Gif-sur-Yvette, France
aymen.boudguiga@cea.fr

Jun Zhang
Telecom ParisTech
Paris, France
jun.zhang@telecom-paristech.fr

Houda Labiod
Telecom ParisTech
Paris, France
houda.labiod@telecom-paristech.fr

Alain Servel
PSA Group
Velizy, France
alain.servel@mpsa.com

Pascal Urien
Telecom ParisTech
Paris, France
pascal.urien@telecom-paristech.fr

ABSTRACT

Connected and automated vehicles aim to improve the comfort and the safety of the driver and passengers. To this end, car manufacturers continually improve actual standardized methods to ensure their customers safety, privacy, and vehicles security. However, these methods do not support fully autonomous vehicles, linkability and confusion threats. To address such gaps, we propose a systematic threat analysis and risk assessment framework, SARA, which comprises an improved threat model, a new attack method/asset map, the involvement of the attacker in the attack tree, and a new driving system observation metric. Finally, we demonstrate its feasibility in assessing risk with two use cases: *Vehicle Tracking* and *Comfortable Emergency Brake Failure*.

CCS CONCEPTS

• **Security and privacy → Security requirements**; **Embedded systems security**; • **Computer systems organization → Embedded and cyber-physical systems**; Dependable and fault-tolerant systems and networks;

KEYWORDS

Automotive Security; Threat Analysis; Risk Assessment; Security Requirements

ACM Reference Format:
Jean-Philippe Monteuuis, Aymen Boudguiga, Jun Zhang, Houda Labiod, Alain Servel, and Pascal Urien. 2018. SARA: Security Automotive Risk Analysis Method. In *CPSS'18: The 4th ACM Cyber-Physical System Security Workshop, June 4, 2018, Incheon, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3198458.3198465

1 INTRODUCTION

Self-driving vehicles such as Tesla model *S* have fostered the development of connected and automated vehicles. Original equipment manufacturers (OEMs) are promoting their first vehicles for 2020.

Automated vehicles sense and monitor their internal and external environment. Indeed, the Automated Driving System (ADS) of a vehicle exploits sensors data and controls the vehicle dynamics to assist the driver through automated features like adaptive cruise control (ACC) [1]. Also, the ADS alerts the driver to potential unsafe road events that require human assistance.

Connected vehicles communicate with other vehicles, road infrastructures, and the Cloud. We refer to communication between vehicles as Vehicle-to-Vehicle (V2V) communication, and between the infrastructures and vehicles as Vehicle-to-Infrastructure (V2I) communication. These communications improve vehicle and passengers safety. For instance, communications with traffic lights enhance the traffic fluidity. V2V communications prioritize vehicles traffic and favor emergency vehicles.

Automated driving system-dedicated vehicles (ADS-DVs) [1] rely on V2X communications and vehicular automation to enhance automated features such as cooperative ACC (Co-ACC). Unlike regular vehicles, ADS-DV might not include human-machine interfaces (e.g., brake pedal, steering wheel…). For instance, in case of blinding attacks on the camera [2], security experts must assess how an ADS can self-detect the traffic light state and pass through the intersection safely.

Indeed, as vehicles support new technologies, threats targeting the ADS increase. At Black Hat 2015, Miller and Valasek [3] demonstrated the hack of a Jeep Cherokee Electronic Control Units (ECU) by exploiting a remote vulnerability on the *Uconnect* head unit. Then, they remotely activated the braking function. That is, this type of security attacks could have resulted in a safety violation by putting at risk human lives.

Facing such emergency, security and safety experts proposed security risk assessment methods without converging to a satisfying solution. A survey [4] of two standardized methods, namely EVITA [5] and TVRA [6], demonstrated some risk computation improvements and converged to a new method taking the advantages from both methods. Despite such effort and recent standards revisions (2016-2017) [1, 7], standards fail to propose a joint methodology.

During standards revisions, new threats emerged [8] questioning current methods applicability. It includes *Malicious observers* threats that link anonymous public data to retrieve confidential data despite confidentiality countermeasures (e.g., certificate pseudonym, anonymous message) [9, 10]. Also, *Altered road infrastructures* [11, 12] threats that send authentic messages with incorrect

CPSS'18, June 4, 2018, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5755-5/18/06…$15.00
https://doi.org/10.1145/3198458.3198465

data elements. Such threats can confuse the vehicle fusion and decision algorithms despite cryptographic countermeasures (e.g., message signature, pseudonym certificate, revocation) [13].

To address these new threats, we review existing methods and highlight their gaps in the case of a driver-less vehicle. Then, we propose:

- a new framework named *SARA* that comprises an improved threat model, a new attack method/asset map, the involvement of the attacker in the attack tree, and a new metric for driver-less vehicles named *Observation*
- a demonstration of our method feasibility through some automotive use cases.

The remainder of this paper is organized as follows. Next section surveys aforementioned methods and positions our method, namely SARA. Section 3 presents SARA framework. Section 4 describes the considered vehicle architecture. Section 5 describes SARA threat analysis process. Section 6 presents SARA risk assessment process and its application to some use cases. Section 7 describes SARA countermeasure process. Section 8 concludes our paper.

2 RELATED WORK

Security risk assessment methods assess the risk of traditional *IT* infrastructures without considering safety implications. Unfortunately, as mentioned, automotive attacks can have safety impacts.

Nowadays, the Society of Automotive Engineers (SAE) defines the security framework of Figure 1, in SAEJ3061 [1]. SAEJ3061 relies on one of the following risk assessment methods: EVITA [5], TVRA 2015 [6], and HEAVENS [14]. Also, SAEJ3061 proposes similar processes with ISO26262 [15] safety risk assessment method to contribute to security risk assessments. However, the nature of this contribution remains unexplained. Our framework clarifies it by integrating the safety expert assignment on safety metrics for safety-related attack goals.

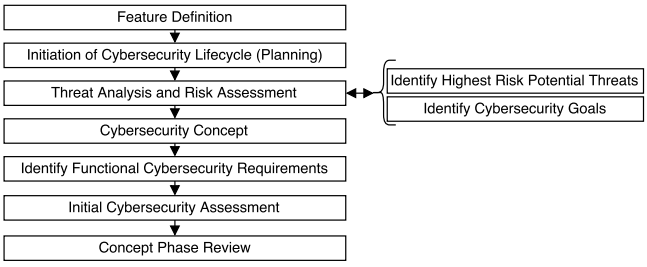

Figure 1: SAE J3061 Concept Phase

In 2009, Henniger et al., [5] proposed EVITA. They defined a risk matrix considering the attack likelihood, the attack severity, and the driver controllability. However, their attack tree definition is unclear. Indeed, the confusion comes from their distinction between attack goals and objectives that are respectively, the roots and the second nodes of their attack tree. Also, they only consider driver control during the risk computation which does not work with driver-less vehicles. To this end, we introduce the *ADS observation and controllability metrics* later to be compliant with SAE standard[1]. Indeed, SAE standard requires the ADS to self-observe potential faults or failures (caused by hazards or threats) to self-control vehicle dynamics and reduce the safety and security risks.

In 2012, Moalla et al., [16] applied TVRA on a connected vehicle. Therefore, they do not consider threats from the internal vehicle network. Besides, their considered threats list is not exhaustive, due to the absence of threat modeling despite standards recommendation [1, 7]. Wolf and Sheibel [17] applied their security risk assessment framework to a generic ECU model. Therefore, the framework suits for subsystems but not for the whole vehicular system. Also, the method does not assess the privacy impact on security risk.

In 2015, Boudguiga et al., [4] proposed a method, named RACE, combining TVRA and EVITA. The authors clarified the definition of EVITA attack tree for automotive experts by using automotive functions instead of EVITA attack objectives. Besides, they proposed a unique risk computation method using EVITA controllability that matches TVRA rating of risk. However, they did not demonstrate RACE feasibility nor the impact of scalable attacks on the computation of risk value.

In 2016, Macher et al., [18] proposed a method named SAHARA. Their method framework just maps attack goals to ISO26262 safety use cases. However, the framework does not allow interactions between security risk and safety metrics. Their method uses the threat model STRIDE [19] which does not consider authentic messages with false data attacks. Also, STRIDE fails to consider attack with multiples security goals. Finally, authors used *DREAD* [20] to assess the security risk. Unfortunately, the discovery of a new attack affects the computation of DREAD. Indeed, if a blog advertises an attack, the value of the metric *discoverability* increases. Then, it increases the values of metrics *reproducibility*, *exploitability*, and *affected users* because, thanks to the leak, an attacker knows how to reproduce and perform the attack massively. That is, DREAD is not suitable for assessing risk. Islam et al., [14] combined STRIDE to Data Flow Diagrams (DFD) to categorize vulnerabilities on an automotive speed limiter. However, as they studied only the speed limiter, their approach does not scale to the whole vehicle system. Dominic et al., [21] proposed a method for autonomous driving systems. As required by standards [7], the authors used attacker profiles to compute the risk value using the metric *Motivation*. However, they only consider surface attacks and not internal attacks such as *ECU confusion attack* [22].

In 2017, the European Telecommunications Standards Institute (ETSI) provides a revised version of TVRA [7]. TVRA relies on industry-proven methods (e.g., Target of Evaluation [23]) and metrics (e.g., attack potential [24, 25]) to assess security risk. Also, TVRA mandates to identify attackers for the computation of risk. However, there is no proposed solution to relate an attacker to its attack. Moreover, TVRA [7] focuses only on telecommunication threats. Therefore, it misses the automation threats domain [8] and ISO26262 *safety* for the risk computation.

Table 10 resumes the related work using multiple criteria based on the mentioned drawbacks: the considered vehicle type, attacker model, threat model, security goal model, attack type, attack model, type of controllability, privacy impact and, safety impact.

Methods do not consider the driver-less vehicle (ADS-DV) as a system of study. This methods always rely on driver control. However, an ADS-DV must rely only on self-control vehicle dynamics

to reduce risk in case of failure from an automated feature [1]. Also, despite standard requirement, many methods do not link the attacker to its attack. Moreover, many methods has insufficient threat-security goal modeling against *vehicle tracking misbehaving nodes* threats [8]. Also, despite having a threat model, methods consider only mono-threat attack instead of multiples threats attacks as mentioned in state of the art [8] due to the lack of attack modeling. Finally, some methods do not consider the impact on safety or privacy despite the European Commission recommendations.

In this paper, we propose SARA, an improved security risk analysis framework for ADS-DV, which comprises safety experts opinions, a new threat model, attack method/asset map, and attack tree definition including the attacker as a metric. Moreover, we define a new metric which considers driver/ADS controllability for the computation of the risk value. Finally, we show SARA feasibility with two uses: *Vehicle Tracking* and *Comfortable Emergency Brake Failure*.

3 SARA FRAMEWORK

SARA framework is organized into four blocks (Figure 2):

- **Feature definition** describes the defense perimeter[1] of the assessed system. The system definition follows two architectures. The *physical* architecture represents interfaces, controllers, sensors, actuators, and communication links. The *logical* architecture represents the data flows issued by aforementioned physical entities. Indeed, an ADS-DV rely on data flows to observe its surrounding environment and control the vehicle dynamics [26]. By knowing the threaten data flows, the expert forecasts the severity of attacks on assets, the capabilities of self-observation, and self-controllability of the automated driving system (ADS).
- **Threat specification** describes SARA *threat to security goal* map, *attack method to asset* map, and SARA *attacker list* definition. The SARA *threat to security goal* map associates our threat model (*STRIDELC*) to our security goal model (*AINCAAUT*)[2]. Then, SARA *attack method to asset* maps a set of assets categories and threats/security goals to an attack method. The latter is a single threat or a set of threats performed by an attacker on an asset. SARA *attackers list* maps an attacker profile and its *attacker capability* score. The latter is the sum of the standardized metrics values (*expertise*, *knowledge*, and *equipment*) required as a minimum to perform an attack[3]. The *attacker capability* serves to compute the *attack likelihood* in the next building block.
- **Risk assessment** returns the risk value of an attack. SARA attack tree defines the attack goal as the tree root and selects its related threats from those identified in the previous threat specification step. Then, we define the *attacker* as the minimally required profile to perform a threat using SARA *attacker list*. Therefore, we compute the attack likelihood of a threat. Then, security and safety experts define attacks goal severity, observation and control values. Finally, experts compute the risk value of an attack goal from the following metrics: severity, observation, controllability and the highest attack likelihood. Section 6 details the risk computation using SARA attack tree.
- **Countermeasures** minimize the computed risk from an attack tree. The applied countermeasures refine the risk level or end the risk assessment process. Indeed, risk analysis is an iterative process that ends once countermeasures have been applied to critical threats until the risk value converges to an acceptable level.

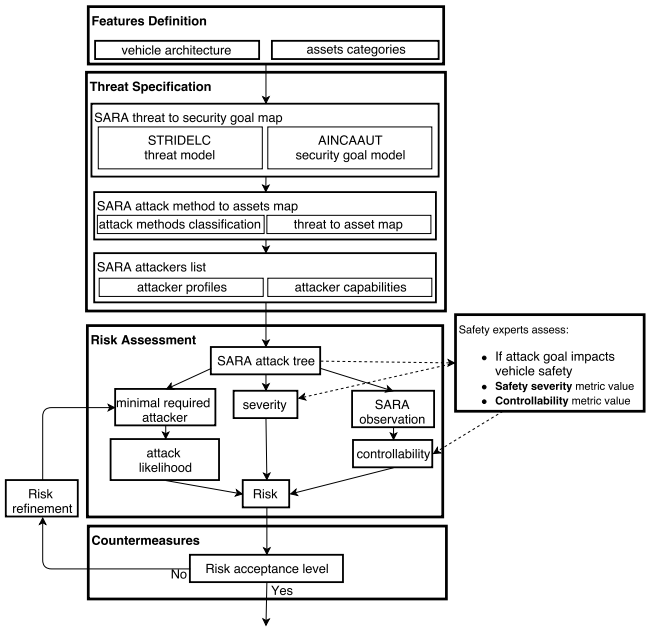

Figure 2: SARA framework

4 AUTONOMOUS VEHICLE ARCHITECTURE

Risk assessment requires security experts to define the evaluated vehicle architecture and its features. The detail level of the system description reflects the architecture maturity and affects risk assessment results. In this section, we present our considered connected and autonomous vehicle architecture.

4.1 Vehicle Physical and Logical Architecture

Our physical and logical vehicle architecture (Figure 3) is based on state of the art disclosed architectures. Our considered architecture is composed of *Electronic Control Unit* (ECU), sensors and actuators connected to each other through several field buses (CAN, FlexRay, Ethernet…). Each ECU achieves an automotive function (i.e., powertrain, infotainment, body, chassis, safety, communication, DAS…) by collecting and processing data from various sources. For instance, sensors (e.g., camera, lidar and radar) sense vehicle internals and its environment to detect mechanical problems, road lines, and traffic signs. ECUs study these sensors perceived information using data fusion and tracking techniques to extract advanced data features (e.g., obstacle class, speed value, localization…). Then, the

[1]Defense perimeter is defined in Target of Evaluation from Common Criteria [23]
[2]STRIDELC and AINCAAUT are detailed later in section 5.1.
[3]Attacker capability metrics are detailed later in section 5.3.1

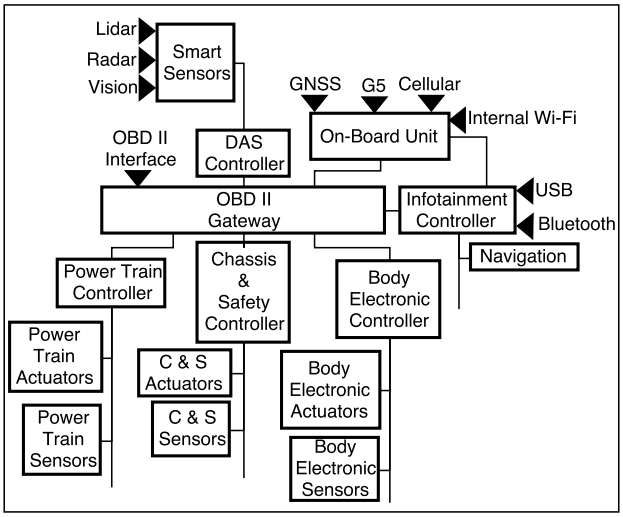

Figure 3: Considered vehicle architecture

DAS Controller (DASC) processes the perceived data into a real-world data model. The latter relies on V2X data collected by the On-Board Unit (OBU) and on the driver inputs via the Infotainment Controller (IC). Once we establish the environment model, DASC improves vehicle driving by ensuring functions such as *Automatic Emergency Braking*, *Automatic Parking*, and *Lane Keeping Assist System*.

4.2 Vehicle Assets categories

Aforementioned components are critical for vehicles control. They are vehicle assets and, therefore, targets of malicious road users. We divide assets into three categories:

- **Equipment** groups ECU, sensors, and actuators (with their installed software and stored data).
- **Data Flow** groups communication (e.g., CAN bus, automotive Ethernet, vehicular communications…) and sensors data flows.
- **External Entity** groups entities interacting with our architecture (e.g., other vehicles, road infrastructures, pedestrians…).

4.3 Vehicle Interfaces

Attackers use vehicles interfaces as entry points to interact with the vehicle internals. Figure 3 shows the following entry points: OBD II interface, USB, Bluetooth, Cellular, G5, GNSS, internal Wi-Fi, and ranged sensors receiver/transmitter (e.g., lidar, radar, camera). Entry points lead to surface attacks performed on the vehicle.

5 THREAT SPECIFICATION

Once the vehicle architecture selected, we identify assets and their related threats using our systematic threats specification. Our method follows three steps: SARA threat/security goal mapping, SARA attack method/asset mapping and SARA attackers list definition.

5.1 SARA *threat to security goal* map

To identify considered threats, we define a new threat model named *STRIDELC* (Table 1). The latter extends STRIDE by adding two categories that are *Linkability* and *Confusion*. The latter refers to the processing of authentic data structure with incorrect content that does not reflect the ground truth state. For instance, a traffic light emits authentic V2X messages with incorrect traffic light states. Therefore, The incorrect state confuses the ADS which must rely on another reliable data source. *Confusion* differentiates from threats such as *Spoofing*, *Tampering*, and *Elevation of Privilege*. Indeed, a source sending authentic messages with incorrect content neither usurps another source identity nor alters a data structure [8]. *Confusion* related security goal is *Trustworthy* which includes countermeasures assessing the trustworthiness of the content and/or its source [27].

Linkability refers to the ability to link pseudonymous or anonymous data to identify the data owner. *Linkability* differentiates from *Confidentiality* as follows. For instance, malicious observers collect vehicle signed cooperative awareness messages (CAMs) on a predefined road path. By tracking vehicle localization contained in the messages, the attacker extract private data such as preferred driving path [2], housing localization, children localization, health status (e.g., hospital, gym, fast-food), and its customers localization. After data processing, the data allow to map to confidential information such as vehicle owner identity using its house localization (despite anonymous/pseudonymous message[9]). The related security goal *Unlinkability* includes countermeasures providing dynamic confidentiality such as pseudonym certificate change scheme [8] or obscuring proxies [28].

For the remaining threat/security goal associations, we refer to Microsoft SDL *STRIDE to security goals* map [29]. Also, we prioritize *Confidentiality*, *Integrity*, *Availability*, and *Trustworthy* for their strong impact on ensuring system safety and as a standard procedure (CIA model). That is, Table 1 depicts the considered threat model and security goal model in SARA.

5.2 SARA *attack method per asset* map

Once the considered threats and security goal of our system of study defined, we map them to system assets.

5.2.1 Mapping threats to asset categories. To this end, we revisit *Microsoft STRIDE-per-Element* map [29]. First, we associate defined assets categories with elements. As mentioned in section 4.2, *Equipment* stores data and processes it using the software. Therefore, we map this asset category to *Data Process* and *Data Store* elements. The remaining associations between asset categories and elements are straightforward. Then, we match our asset categories to our *STRIDELC* threat model. As defined in Table 1, only data emitters such as *Equipment* and *External Entity* produce authentic messages with incorrect content. Therefore, we map *Confusion* threat to *Equipment* and *External Entity* asset categories. *Linkability* threat targets data related to our system of definition. This includes both *Data Store* and *Data Flow* which are, if not confidential, public or semi-public information (e.g., logs) potentially used to gather confidential information as mentioned. Finally, Table 2 maps STRIDELC threats to our defined assets categories.

Table 1: SARA Threat-Security goal Models

STRIDELC Threats categories	Explanation	**AINCAAUT** security goals[a]
Spoofing	impersonate someone or something else	**A**uthenticity
Tampering	to modify data or functions	**I**ntegrity (*)
Repudiation	cannot traced back the author actions	**N**on-Repudiation
Information Disclosure	to access to confidential data	**C**onfidentiality(*), (Privacy)
Denial of Service	interrupt a system legitimate operation	**A**vailability (*)
Elevation of Privilege	perform unauthorized actions	**A**uthorization
Linkability [22]	deduce the owner identity from owner public unidentified data	**U**nlinkability [28], (Privacy)
Confusion [22]	a data source confuses the system by sending incorrect data within authentic data structure	**T**rustworthy(*) [8]

[a](*) identifies prioritized goal

Table 2: STRIDELC-per-asset categories map

Element [29]	Assets (Section 4.2)	STRIDELC threats (Table 1)							
		S	T	R	I	D	E	L	C
External Entity	External Entity	✓		✓					✓
Data Process	Equipment	✓	✓	✓	✓	✓	✓	✓	✓
Data Store									
Data Flow	Data Flow		✓		✓	✓		✓	

5.2.2 Defining attack methods classes. Once the threats-assets map defined, we need to define the *attack methods classes*. Indeed, as mentioned in standards [7, 17], an attack method groups one or multiple threat categories. For instance, an attack *jamming vehicular communication channel* targets the communication channel using a single threat category named *Denial of Service*. On the contrary, an attack *greedy jamming vehicular communication channel* includes an *Elevation of Privilege* threat and a *Denial of Service* threat. Indeed, a malicious vehicle can saturate the network using its elevated status to consume more channel bandwidth than other vehicles limiting their access to channel bandwidth [16]. As defined in section 5.1, we prioritize the bandwidth *availability* as a security goal over the removal/modification of attacker privileges (*Authorization*) to avoid constant attacks.

To define attack method, we use *CIA* model and TVRA *Threat Tree* [7]. To impact the system of study, we assume that attack methods must target one of CIA security goals that are *Confidentiality*, *Integrity*, and *Availability*. TVRA *Threat Tree* validates this assumption by grouping threats into four groups: *Interception*, *Manipulation*, *Denial Of Service*, and *Repudiation*. However, we disagree with this approach. Indeed, an attacker must launch an action *Interception*, *Manipulation*, *Denial Of Service* to be able to repudiate his action. Therefore, *Repudiation* threat cannot be an attack method itself. Therefore to match SARA prioritized security goals, we define four attacks method classes as follows:

- **A***lter* attacks aim to modify data which relate to *Tampering* threats/*Integrity* security goals
- **L***isten* attacks aim to monitor data which relate to *Information Disclosure* threats/*Confidentiality* security goals
- **D***isable* attacks aim to deny access to data which relate to *Denial of Service* threats/*Availability* security goals
- **F***orge* attacks aim to create incorrect data which relate to *Confusing* threats/*Trustworthy* security goals

5.2.3 Mapping attack method classes/asset categories. Once the attack method classes defined, we map the major threat of each attack method (e.g., Tampering) to our *STRIDELC-per-asset categories map* (Table 2). As a result, we obtain the *SARA attack method per asset map* (Table 2). This map allows a systematic tool to map multiple threats/security objectives to assets which can be useful to build attack tree, Petri-nets or graphs. Also, non-security experts can use *SARA attack method per asset map* to avoid security goal omission/misidentification. For instance, in[30], the author identifies a spoofing threat from a malicious diagnostic tester as an *Authorization* security goal whereas it is an *Authenticity* security goal.

5.3 SARA attackers list

As mentioned in standards[7, 23], an attacker, its actions, and its targeted asset define an attack. To this end, this section defines SARA attackers list and evaluates it.

5.3.1 SARA attackers profiles definition. We define an attacker as the combination of an attacker profile and an attacker capability. To define the attacker profile, we refer to previous methods [21, 22] and the attacker model defined in [8].

Table 4 depicts a list of attackers (A) and their corresponding capabilities (Ca_A). Ca_A depends on multiple standardized factors [24, 25]:

- **Knowledge factor** (K) refers to the attacker knowledge regarding the chosen system. K can be public, restricted, sensitive and critical.
- **Expertise factor** (Ex) refers to an attacker expertise. It specifies four attacker categories. A layman is a person without specific security knowledge. A proficient is a person with basic security knowledge. An expert has a strong security culture learned from his past hacks and attended conferences. Multiple experts are a group of experts united around a common attack goal. Multiple experts launch simultaneous attacks to achieve their attack goal.

Table 3: Mapping SARA attacks method classes to assets categories

attack method classes	Alter	Listen	Disable	Forge
AINCAAUT prioritized (*) security goals (Table 1)	Integrity(*) Non-Repudiation Authorization	Confidentiality(*) Non-Repudiation Authorization Unlinkability	Availability(*) Non-Repudiation Authorization	Authenticity Non-Repudiation Authorization Trustworthy(*)
SARA asset categories (Section 4.2)	Data Flow / Equipment	Data Flow / Equipment	Data Flow / Equipment	Ext. Entity / Equipment

Table 4: ISO metrics to Attackers Capabilities

SARA	ISO metrics [24, 25]			SARA
Attacker Profiles (A)	Expertise (Ex)	Knowledge (K)	Equipment (Eq)	Ca_A
Thief, Mr.Nobody	Layman (0)	Public (0)	Standard (0)	0
Researchers	Experts (8)	Public (0)	Specialized (4)	12
Evil Mechanic	Expert (6)	Restricted (3)	Specialized (4)	13
Organized Crime	Proficient (2)	Sensitive (10)	Specialized (4)	16
Hacktivist	Experts (8)	Sensitive (4)	Multi-bespoke (9)	21
Foreign Government	Experts (8)	Critical (11)	Multi-bespoke (9)	28

- **Equipment factor** (Eq) refers to the equipment needed by an attacker to perform an attack. There are 4 types of equipment. A standard equipment is an equipment already available for the attacker. A specialized equipment needs to be ordered from a specialized shop. A bespoke equipment is not easy to purchase and is expensive to create. Some attacks require multiple bespoke pieces of equipment which are hardly available and very expensive.

Security expert sums factor values (C_j) to compute the capability Ca_A of an attacker A as follows:

$$Ca_A = \sum_{j \in \{K, Ex, Eq\}} C_j \quad (1)$$

5.3.2 SARA attacker profile evaluation. This approach differs from previous work in two ways. First, an attacker profile defines the attacker capability whereas previous work defined attacker capability based on threats. Second, we optimize previous approaches by reducing the total decision time and the total number of choices combinations. The latter is the number of possible combinations proposed to the expert to evaluate the attacker capability. Our method proposes 7 possible combinations (7 attacker profiles) to evaluate the attacker capability whereas the standard offers 48 combinations. Therefore, our method reduces the decision time. Indeed, if we assume the same time of evaluation per metric(t), an expert needs only a single t to evaluate the attacker capability instead of three t. That is, SARA attacker profiles optimize the computation of attacker capability.

Table 5: Decision-Gain regarding Choices and Time

Metric j ∈ {K, Ex, Eq, A}		K	Ex	Eq	A
Available choices per metric	n_j	4	4	4	7
Setting time per metric	t_j	t	t	t	t
Total of choices combination	$\prod_j n_j$	48			7
Total setting time	$\sum_j t_j$	$3 \times t$			t

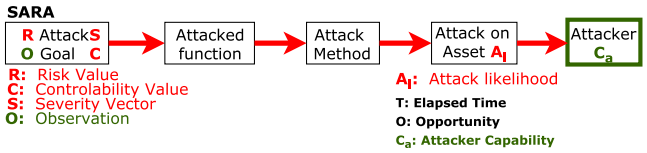

Figure 4: SARA attack tree

6 RISK ASSESSMENT

This section presents SARA risk assessment. We first define SARA attack tree and its metrics used for risk computation. Then, we apply our risk assessment method to two use cases.

6.1 SARA attack tree

An attack tree defines threats used by attackers to reach an attacking goal (Figure 4). Attackers reach their goal through attacked automotive functions. *Attacked functions* simplify targeted components identification within the vehicle and clarify the attack description for automotive experts. Then, SARA *attack method* (Section5.2) maps an attack method the impacted assets using SARA *attack method to asset* map (Table 3). Finally, we associate a minimally required *attacker* (Table 4) to the *attack on an asset* which maps one or multiple threats to the impacted asset(Table 3). Experts compute the attack goal risk score based on the highest attack likelihood score among all attacked assets.

6.2 Attacker profile and attack likelihood

SARA attacker profiles help experts to identify the minimally required attackers regarding an attack goal. As mentioned, an attack is composed of one or multiples threats which are performed by

Table 6: Standardized Mapping Attack Likelihood [5, 7, 31]

AP_A	Description	AI
[0, 9]	Basic	5
[10, 13]	Enhanced basic	4
[14, 19]	Moderate	3
[20, 24]	High	2
> 24	Beyond high	1

attackers. Therefore, as mentioned, the success of an attack depends on the attacker capability but also on the elapsed time (T) and on the required opportunity (WO) to perform the attack [5]. The time factor is the time needed to identify and successfully realize an attack considering the attacker capability. The opportunity tells if an attack requires a special window of opportunity to be executed or it can be easily executed.

Experts compute attack potential (AP_A) using the values of *attacker capability* Ca_A, normalized *elapsed time* T and *opportunity* metrics WO as follows:

$$AP_A = Ca_A + T + WO \quad (2)$$

That is, attacks requiring the lowest minimal attack potential are more likely to occur (Table 6). As a result of improving attacker capability, we reduce the total metrics for attack potential from initially 5 to 3.

6.3 Attack goal severity

Standardized severity factors are safety (S_s), privacy (S_p), financial (S_f) and operational (S_o) [31]. SARA severity relies on the previous factors values and expert motivation for severity vector computation (Table 8):

$$\vec{S} = (S_s, S_p, S_f, S_o) \quad (3)$$

We choose a maximization approach by assuming that all severity factors have equal importance. That is, SARA severity value is the highest severity vector coefficient. For instance, we consider as attack goal *an unauthorized braking from one vehicle at low speed* with specific severity vector (e.g., S = (1, 1, 0, 2)). The maximized severity value is $S = S_o = 2$. Therefore, we reduce risk assessment time by avoiding the full risk vector computation. However, our approach still supports vector approach if threat risk must be evaluated for each severity factor separately. SARA severity considers attack goal scalability. For instance, if the aforementioned attack goal occurs in a traffic jam. An unauthorized brake has a strong impact on multiple vehicles safety and their operational state. The severity of this situation (S = 3) is higher than in the single-vehicle case (S = 2). That is, SARA Severity is flexible (e.g., maximization or vector approach), supports severity absence (e.g., S = 0) and is scalable (e.g., single or multiple attacks).

6.4 Attack goal observation and controllability

System control requires system internal and external observation to anticipate system failures caused by hazards or threats. The fully autonomous vehicle cannot rely on human perception. We tackle this issue with a new metric called *Observation* (O). The latter defines system tolerant default and its ability to detect errors and faults. Therefore, it controls system security risks. *Observation* has two values: perceptible (O = 1) and imperceptible (O = 0). Figure 5 illustrates the use of *Observation* in practice.

Table 7: Observation and controllability classification

O	C	Meaning
1	0	ADS observation is available but no accident avoidance is required
	1	ADS observation is available and accident avoidance is required using ADS response
0	2	ADS observation is unavailable/uncertain, driver response is required
	3	ADS observation is unavailable/uncertain, driver response is impossible/unavailable

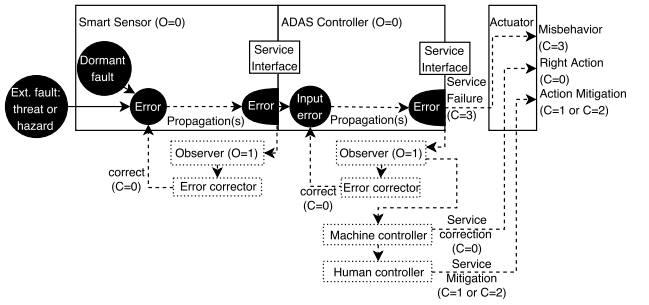

Figure 5: Concept of Observation and Controllability

A mechanic attacker targets a sensor connected to a vehicle by altering a sensor calibration. The faulty sensor creates system error and probably a system failure. At initialization, expert considers threat observation as null. However, with appropriate countermeasures, the threat becomes perceptible which leads to risk reduction. The metric *Observation* advantages are considering vehicle safety without human control and forecasting vehicle architecture countermeasures to failures (Table 7). Architecture countermeasures control fault propagation and rely for example, on data redundancy, watchdog or IDS. Note that data redundancy increases vehicle cost. *Controllability* (C) quantifies the autonomous system or driver influence on security risk [32]. C ranges from 0 to 3: 3 refers to the absence of driver/ADS controllability over the vehicle, whereas 0 is the opposite (Table 7).

6.5 Risk computation

SARA computes the risk score using the SARA matrix function (f) defined in Table 9.

$$R = f(C, S, AI) \quad (4)$$

The risk score ranges from insignificant (R0) to unacceptable (R7+). Expert uses risk score to evaluate a threat and decide if countermeasures are needed. Besides considering machine controllability, our approach advantage is to rely on the same matrix for safety and none safety-related use cases. Also, it is similar to ASIL computation method which reduces the gap between security and safety.

[4] refer to Table 7
[5] refer to Table 8

Table 8: SARA Severity

S	Safety	Privacy	Financial	Operational
0	No injuries	undisclosed or unlinkable data	No loss	No impact on vehicle performance
1	single light to moderate injury	one identified vehicle	> 100$	one small impact on a vehicle
2	single severe injury or multiples moderates injuries	one vehicle tracking or identification of multiple vehicles	> 1000$	one big impact or many small impacts
3	single life threatening injury or multiple severe injuries	multiple vehicles tracking	> 10000$	big impact on many vehicles

Table 9: SARA Risk Matrix (C: Controllability, S: Severity, Al: Attack Likelihood)

C [4]	S [5]	Al [6]				
		1	2	3	4	5
0	1	R0	R0	R1	R2	R3
	2	R0	R1	R2	R3	R4
	3	R2	R3	R4	R5	R6
1	1	R0	R1	R2	R3	R4
	2	R1	R2	R3	R4	R5
	3	R3	R4	R5	R6	R7+
2	1	R1	R2	R3	R4	R5
	2	R2	R3	R4	R5	R6
	3	R4	R5	R6	R7	R7+
3	1	R2	R3	R4	R5	R6
	2	R3	R4	R5	R6	R7
	3	R5	R7	R7	R7+	R7+

6.6 SARA risk assessment applications

In this section, we assess the security risk of two cases of study: the *Vehicle Tracking* [22] and the *Comfortable Emergency Brake Failure* [11].

6.7 Vehicle Tracking

A vehicle broadcasts periodically signed *cooperative awareness messages* (CAMs) in a defined area. The latter contains public anonymous/pseudonymous data related to the vehicle localization. However, despite anonymous data, an observer eavesdrops and tracks vehicles messages. Then, knowing vehicles positions history in the neighborhood, an observer maps the pseudonym certificate of the car owner to its house address and so, its identity. A global observer can track information on the supply chain of a company such as the supply chain path or the position and or clients localization/identity. Indeed, The observer reconstructs the vehicle path history using the position data in its CAM. Also, the observer checks where the vehicle stopped, then maps the localization to potential customers address. Robbers could check the presence of police vehicles in the area visually or through the rights of their pseudonym certificate then tracking them [2]. That is, the privacy impact is high. Figure 6 depicts an attack tree of the scenario based on SARA framework (Figure 2). First, we initialize the attack goal settings (Figure 4) as follow.

[6]refer to Table 6, if S=0, it means an absence of risk.

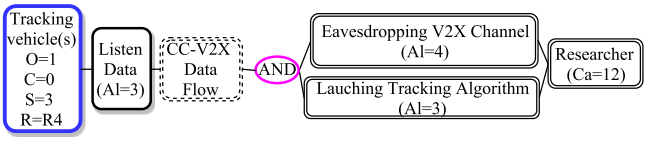

Figure 6: Attack tree of Vehicle Tracking

- The severity factor concerns only privacy as the attack goal does not affect safety, financial or operational aspects of the system. Also, the attack allows the tracking of multiple vehicles. That is, the Severity factor value is $S_p = 3$ (Table 8).
- The attack does not impact the ADS observation operational state (O = 1, (Table 7).
- Therefore, the attack does not require any control over the vehicle dynamic (C = 0, Table 7).

Second, using the attack tree goal and SARA *attack method/asset map* (Table 3), we define the potential attack methods, threatened asset category, possible threats, and the minimum required attacker.

- the attack method class is *Listen* because the attack goal requires to monitor vehicular communication.
- the threaten asset category is *Data Flow* because the attack focus on the data elements contained in CAMs.
- the identified security goals are *Confidentiality*, *Unlinkability*, *Authorization*, and *Non-Repudiation*.
- the minimum required attacker is *Researchers* [2].

Based on the identified security goals and the *SARA threat to security map* (Table 1), we identify the following threats. To achieve its attack goal, the attacker eavesdrops the communication channel to "read" the message content. *Confidentiality* is the issue due to disclosed message content. The second threat concerns *Unlinkability* because the attacker can extract private information from public data using tracking algorithms [2, 9, 27] as aforementioned. In this case, threats against *Authorization* and *Non-Repudiation* are not part of the scope. Indeed, vehicles broadcast their messages that are accessible to everyone in the transmission range.

Third, we compute the attack likelihood value of *Eavesdropping on the V2X channel* and *Tracking CAMs* threats by assessing the following metrics (Equation 2).

- *attacker capability* value (Ca),
- *Elapsed Time* value (T), and
- *Window of Opportunity* value (WO).

As defined in [2], the attacker profile is a *Researcher* which *attacker capability* value ($(Ca_{researcher})$) is 12 (Table 4). To perform *Eavesdropping on the V2X channel*, we assume an attacker requires less

than a day. Therefore, referring to *Elapsed Time* map [5, 7], the *Elapsed Time* value is 4 (T = 4). *Tracking CAMs* requires 2 weeks [2]. Therefore, the *Elapsed Time* value is 4 (T = 4). The attacker does not need a *Window of Opportunity* to reach his attack goal. Therefore, the *Window of Opportunity* value for both threats is null (WO = 0) based on [5, 7]). Then, using Equation 2, we compute *Eavesdropping on the V2X channel* and *Tracking CAMs* attack potential values (respectively, AP = 0 and AP = 0). Finally, using Table 6, we map their attack likelihood values (respectively, AI = 4 and AI = 3).

Fourth, we compute the attack likelihood value of the attack method class *Listen* using:

- each threat attack likelihood value, and
- if multiple threats, the logical operator definition.

Tracking CAMs is possible only if the attacker is *Eavesdropping on the V2X channel*. Therefore, the logical operator is *AND*. Using the threats attack likelihood value and the logical operator definition, we compute *Listen* attack likelihood value (AI = 3).

Fifth, we compute the security risk value on the attack goal using:

- the attack likelihood value of each attack method, and
- if multiple attack methods, the logical operator definition.
- Controllability and Severity values defined during the attack goal setting.

Listen is the only attack method for this attack goal. Using Equation 4 and the risk matrix (Table 9, we compute the risk value of the attack goal *Tracking vehicles* (R = R4).

Although not high, SARA risk value is pertinent regarding current work. As ongoing researches on the autonomous vehicle enhance tracking algorithms, the price decrease and the accessibility increase of tracking device will increase the spectrum of attackers and therefore the attack likelihood over time. Countering such attack remains difficult. Indeed, the removal of the identifier from V2X messages may increase the identification process and favors spoofing attacks. On the other hand, the removal of data elements from V2X messages threatens cooperative awareness applications that rely on both classification and location data from the lidar and the CAM to detect accurately pedestrian [26]. Even though countermeasures such as *pseudonym change strategies* exist, their efficiency still need to be evaluated [27, 33].

This analysis confirms two facts. First, the assessed risk score reflects the current situation regarding this attack. No satisfying solution has been proposed yet despite mutual efforts from standardization and industrials. Second, the need to revise or extend current threat models such as STRIDE. As we see, privacy is not just a matter of confidentiality or anonymity but of unlinkability of public data which may variate following the needs of cooperative awareness applications.

6.8 Comfortable Emergency Brake Failure

To assess the impact of a faulty traffic light on a driver-less vehicle, we assess the risk of an attacked automated and connected feature called *Comfortable Emergency Brake* feature (CEB) [34].

This feature uses the content of the Signal Phase and Timing (SPaT) and of the MAP messages emitted from a connected traffic light. Then, once the traffic light is in the camera line of sight, the automated driving system (ADS) collects, processes the output of the camera and the V2X messages. The ADS compares the state of the traffic light inside the SPaT message with the camera output for color matching. The camera output assess the correctness of the SPaT before braking.

Due to the lack of security mechanisms [34], we assess the security risk of this feature. A potential attack goal is *CEB fails to trigger braking at a red light*. We define the following settings:

- The severity factor concerns mainly safety. Therefore, the security experts need to discuss the safety impacts with the safety experts to assess the following metrics. We assume in this case the chosen severity metric value is 3(S_p = 3, Table 8).
- The attack impacts the system observation (O = 0).
- There is no driver response with a driver-less vehicle (C = 3).

We assume the ADAS controller is the automotive function that processes the data and decides to brake. Therefore, the attacked automotive function is ADAS.

To reach their attack goal, attackers must send a color different than red to the vehicle driving system. Based on SARA *attack method classification*, three attack classes fulfill such conditions: *Alter*, *Disable*, *Forge*. Figure 8 depicts the potentially obtained attack tree.

For the sake of clarity, we will discuss on the most interesting threats:

First, we discuss about attacks disabling the optical flow are efficient. They require no effort from the attacker and hardly detectable by the system. Indeed, a delivery van standing in front a traffic light is hardly seen as an attack. Also, physical attacks on the road infrastructure are hardly detectable. Indeed, Figure 7a depicts a physically damaged traffic light. Despite targeting the infrastructure, this attack has an impact on the vehicle waiting for the red light to turn green. Moreover, this attack is easily scalable for an attacker which means the system cannot rely on other surrounding traffic lights. Even in a big city such as Paris, it took one month to report and repair the damaged infrastructure.

We assume most vehicular communications to be cryptographically secured as requested by standards [13]. Therefore, we do not focus on attacks altering the content of the SPAT message using MITM.

Next, we discuss attacks forging data. However, as mentioned, road infrastructures are easily accessible and can misbehave. Indeed, an altered traffic light [11] can emit a SPAT with an incorrect red state instead of a ground truth green state. If an attacker physically damage the traffic light (Figure 7a) or blind the camera [8], the driving system can only rely on an incorrect SPaT. If the SPAT emits incorrect green state, the vehicle can cross the intersection without seeing unconnected objects coming from its left or its rights leading to a potential collision. This threats goal is to confuse the system [22] and require appropriate countermeasure to allow the automated driving system to take safe decision. The present analysis shows that the threats impacting the vehicle are not always vehicle-oriented but infrastructure-oriented. Indeed, road infrastructures are not all connected and render cryptographic approaches useless (Figure 7b). Although some work attacked the camera using faulty road Sign [12] or faulty traffic light [11], none of them assessed the security risk on ADS-DV/driver-less vehicle.

(a) Traffic Light (b) Road Sign

Figure 7: Faulty Road Infrastructures

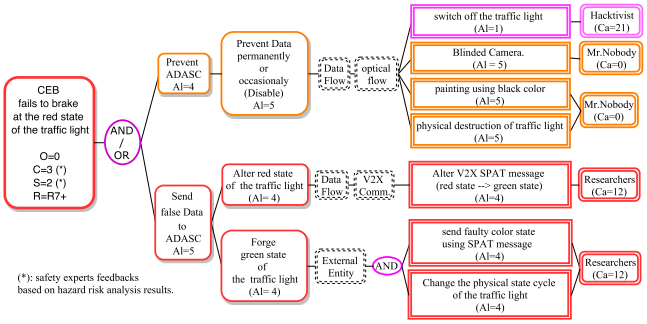

Figure 8: A. Tree for Comfortable Emergency Brake Failure

7 COUNTERMEASURES

SARA final step is to apply countermeasures to reduce highest attack risk values. Then, we re-iterate SARA risk assessment application process until reaching an acceptable risk value. We initially reduce the reiteration process by setting an acceptable risk value for each attack goal. The setting of R and S values define the maximal accepted attack likelihood (AI_{wanted}) for all attacks on asset related to that attack goal. Finally, we apply countermeasures on attacks on asset until all their attack likelihood values (AI) verify:

$$AI \leqslant AI_{wanted} \quad (5)$$

For instance, in the case of the *Comfortable Emergency Brake Failure* (Figure 8), we set a risk value of *R5* as a satisfying requirement without changing *Severity* and *Controllability* values. Then, we compute Al_{wanted} using *SARA Risk Matrix* (Table 9). The wanted attack likelihood value is 1. Therefore, we know that we need to assess all the threats with an attack likelihood value greater than 1. Doing so, we do not re-iterate SARA risk assessment application process and we know which threats require to be countered first.

8 CONCLUSION

This paper presents a survey of existing threat and risk analysis methods and a new security risk analysis for future automated driving system-dedicated vehicles. SARA provides a framework towards safety experts involvement in security processes. Therefore, this paper highlights the need of methods for proper threat analysis coverage against human omissions in order to consider recent concerns regarding the trustworthiness and privacy of the driver-less vehicle. Also, this paper proposes some improvements to existing standards. Finally, SARA proposes a new metric for attack observation for DAS controllability. Indeed, automated driving system-dedicated vehicles can be designed without having human interaction and therefore must be able to detect an attack in order to control and reduce risk value. To this end, we presented the potential impact of a malicious observer and faulty road infrastructures on the vehicle. Our future work consists of extending this work with a data-centric approach.

ACKNOWLEDGMENTS

This work was supported by a PhD Grant from French ANRT (Association Nationale de la Recherche et de la Technologie). The authors thank Benjamin Venelle and Jonathan Petit for their insightful comments and suggestions.

Table 10: Summary of State of Art

Related Work	Type of Vehicle	Attacker Model (required[7])	Threat Model regarding [8, 22]	Security Goals Model	Attack type	Attack Modeling	Attack Scalability	Vehicle Controllability metric (required [1])	Privacy Impact	Safety Impact
EVITA [5]	Automated, Connected	Missing	STRIDE (Insufficient)	Missing Trustworthy Unlinkability	multi-threats	Attack Tree	Severity	Driver Control	EVITA Severity	EVITA Severity
Wolf et al., [17]	Automated	Missing	STRIDE (Insufficient)	AINCAA (Insufficient)	multi-threats	Attack Tree	Severity	Missing	Missing	Severity
Moalla et al., [16]	Connected	Missing	Threats List (Limited)	Missing	mono-threat	Missing	TVRA Impact	Missing	Missing	Missing
SAHARA [18, 35]	Automated	Missing	STRIDE (Insufficient)	AINCAA (Insufficient)	mono-threat	Missing	R-metric (DREAD)	Driver Control	Missing	ASIL Evaluation
RACE [4]	Automated, Connected	Missing	Threats List (Limited)	Missing	multi-threats	Attack Tree	old Impact (TVRA [6])	Driver Control	EVITA Severity	EVITA Severity
Dominic et al., [21]	Automated, Connected	random attackers list	STRIDE (Insufficient)	AINCAA (Insufficient)	mono-threat	Threat Matrix	Missing	Missing	Impact Level	Impact Level
HEAVEN [14]	Automated	Missing	STRIDE (Insufficient)	AINCAA (Insufficient)	multi-threats	Data Flow Diagram	Impact Level	Missing	Impact Level	Impact Level (Severity)
DEWI [30]	Connected	Missing	STRIDE (Insufficient)	AINCAA (Insufficient)	mono-threat	Missing	Missing	Missing	Missing	Missing
Schmittner et al., [36]	Automated, Connected	random attackers list	TID (Insufficient)	ICA (CIA model) (Insufficient)	mono-threat	Missing	SAE Severity [31]	Driver Control	SAE Severity [31]	SAE Severity [31]
TVRA 2017 [7]	Connected	random attackers list	STRID (Insufficient)	AINCA (Insufficient)	multi-threats	Threat Tree	Missing	Missing	Missing	Missing
SARA	Automated, Connected, ADS-DV	attackers metric (Section 6.2)	STRIDELC	AINCAAUT model	multi-threats	Attack Map (ALDF) + Attack Tree with attacker	Revisited SAE Severity	Automated Driving System Control	Revisited SAE Severity [31]	Revisited SAE Severity [31]

REFERENCES

[1] SAE On-Road Automated Vehicle Standards Committee and others. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, 2016.

[2] Jonathan Petit, Djurrre Broekhuis, Michael Feiri, and Frank Kargl. Connected vehicles: Surveillance threat and mitigation. *Black Hat Europe*, 11:2015, 2015.

[3] Charlie Miller and Chris Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015, 2015.

[4] Aymen Boudguiga, Antoine Boulanger, Pascal Chiron, Witold Klaudel, Houda Labiod, and Jean-Christophe Seguy. Race: Risk analysis for cooperative engines. In *New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on*, pages 1–5. IEEE, 2015.

[5] Olaf Henniger, Ludovic Apvrille, Andreas Fuchs, Yves Roudier, Alastair Ruddle, and Benjamin Weyl. Security requirements for automotive on-board networks. In *Intelligent Transport Systems Telecommunications,(ITST), 2009 9th International Conference on*, pages 641–646. IEEE, 2009.

[6] TS ETSI. 102 165-1:" telecommunications and internet converged services and protocols for advanced networking (tispan). *Methods and protocols*, pages 2011–03, 2011.

[7] TS ETSI. 102 165-1:" telecommunications and internet converged services and protocols for advanced networking (tispan). *Methods and protocols*, pages 2017–10.

[8] Jonathan Petit and Steven E Shladover. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):546–556, 2015.

[9] Björn Wiedersheim, Zhendong Ma, Frank Kargl, and Panos Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, pages 176–183. IEEE, 2010.

[10] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11:2015, 2015.

[11] Zhiyi Li, Dong Jin, Christopher Hannon, Mohammad Shahidehpour, and Jianhui Wang. Assessing and mitigating cybersecurity risks of traffic light systems in smart cities. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):60–69, 2016.

[12] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017.

[13] Ieee standard for wireless access in vehicular environments–security services for applications and management messages - amendment 1. *IEEE Std 1609.2a-2017 (Amendment to IEEE Std 1609.2-2016)*, pages 1–123, Oct 2017.

[14] Mafijul Md Islam, Aljoscha Lautenbach, Christian Sandberg, and Tomas Olovsson. A risk assessment framework for automotive embedded systems. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, pages 3–14. ACM, 2016.

[15] ISO26262 ISO. 26262: Road vehicles-functional safety. *International Standard ISO/FDIS*, 26262, 2011.

[16] Rim Moalla, Houda Labiod, Brigitte Lonc, and Noemie Simoni. Risk analysis study of its communication architecture. In *Network of the Future (NOF), 2012 Third International Conference on the*, pages 1–5. IEEE, 2012.

[17] Marko Wolf and Michael Scheibel. A systematic approach to a qualified security risk analysis for vehicular it systems. In *Automotive-Safety & Security*, pages 195–210, 2012.

[18] Georg Macher, Eric Armengaud, Eugen Brenner, and Christian Kreiner. Threat and risk assessment methodologies in the automotive domain. *Procedia computer science*, 83:1288–1294, 2016.

[19] Michael Howard and David LeBlanc. The stride threat model. from the book writing secure code, 2002.

[20] David LeBlanc and Michael Howard. *Writing secure code*. Pearson Education, 2002.

[21] Derrick Dominic, Sumeet Chhawri, Ryan M Eustice, Di Ma, and André Weimerskirch. Risk assessment for cooperative automated driving. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pages 47–58. ACM, 2016.

[22] Jonathan Petit, Michael Feiri, and Frank Kargl. Revisiting attacker model for smart vehicles. In *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*, pages 1–5. IEEE, 2014.

[23] Common Criteria. Common methodology for information technology security evaluation, evaluation methodology, v3.1, revision 5. *Common Criteria*, 2017.

[24] Information Technology - Security Techniques - Methodology for IT Security Evaluation. Standard, International Organization for Standardization, Geneva, CH, august 2008.

[25] Information technology – Security techniques – Evaluation criteria for IT security. Standard, International Organization for Standardization, Geneva, CH, august 2009.

[26] Pierre Merdrignac, Oyunchimeg Shagdar, and Fawzi Nashashibi. Fusion of perception and v2p communication systems for the safety of vulnerable road users. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1740–1751, 2017.

[27] Norbert Bismeyer, Sebastian Mauthofer, Kpatcha M Bayarou, and Frank Kargl. Assessment of node trustworthiness in vanets using data plausibility checks with particle filters. In *Vehicular Networking Conference (VNC), 2012 IEEE*, pages 78–85. IEEE, 2012.

[28] William Whyte, André Weimerskirch, Virendra Kumar, and Thorsten Hehn. A security credential management system for v2v communications. In *Vehicular Networking Conference (VNC), 2013 IEEE*, pages 1–8. IEEE, 2013.

[29] Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. Threat modeling-uncover security design flaws using the stride approach. *MSDN Magazine-Louisville*, pages 68–75, 2006.

[30] Marco Steger, Michael Karner, Joachim Hillebrand, Werner Rom, and Kay Römer. A security metric for structured security analysis of cyber-physical systems supporting sae j3061. In *Modelling, Analysis, and Control of Complex CPS (CPS Data), 2016 2nd International Workshop on*, pages 1–6. IEEE, 2016.

[31] SAE International. Cybersecurity guidebook for cyber-physical vehicle systems. Standard, SAE International, March 2016.

[32] ISO. Road vehicles – Functional safety, 2011.

[33] Attila Jaeger, Norbert Bißmeyer, Hagen Stübing, and Sorin A Huss. A novel framework for efficient mobility data verification in vehicular ad-hoc networks. *International Journal of Intelligent Transportation Systems Research*, 10(1):11–21, 2012.

[34] Zaydounr Y Rawashdeh, Trong-Duy Nguyen, Anoop Pottammal, and Rajesh Malhan. Comfortable automated emergency brake for urban traffic light based on dsrc and on-board sensors. Technical report, SAE Technical Paper, 2017.

[35] Georg Macher, Harald Sporer, Reinhard Berlach, Eric Armengaud, and Christian Kreiner. Sahara: a security-aware hazard and risk analysis method. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, pages 621–624. IEEE, 2015.

[36] Christoph Schmittner, Zhendong Ma, Carolina Reyes, Oliver Dillinger, and Peter Puschner. Using sae j3061 for automotive security requirement engineering. In *International Conference on Computer Safety, Reliability, and Security*, pages 157–170. Springer, 2016.