Secure by Design Autonomous Emergency Braking Systems in Accordance with ISO 21434

Adriana Berdich and Bogdan Groza

Abstract With the ever-increasing degree of inter-connectivity inside vehicles and the emergence of self-driving capabilities, security has become a critical demand since, without proper consideration, adversaries may become capable to remotely control vehicles endangering the life of occupants and bystanders. While several cybersecurity standards have recently emerged, such as the ISO 21434, the secure design of various automotive components is still challenging. In this chapter we make an in-depth analysis of the Autonomous Emergency Braking (AEB) system, i.e., a system designed to avoid collisions between the car and objects in front of it, having security objectives in mind. We make a careful evaluation of adversarial actions, that is, the manipulation of various sensor data and commands that are sent over CAN buses and we follow the ISO 21434 to reach concrete cyber-security goals regarding the system. We account for various types of attacks, ranging from the more conspicuous fuzzy or DoS attacks, to less visible stealthy attacks that induce small biases in the system to evade detection.

Key words: automotive security, control system, intrusion detection, intrusion prevention, autonomous emergency braking, ISO 21434

1 Introduction

In the past century, vehicles mediated a dramatic change of our ecosystem, not only by allowing us to safely travel over great distances but also by allowing us to

Adriana Berdich
Faculty of Automatics and Computers, Politehnica University of Timisoara, Romania, e-mail: adriana.berdich@aut.upt.ro, Phone +40-256-403-242

Bogdan Groza
Faculty of Automatics and Computers, Politehnica University of Timisoara, Romania, e-mail: bogdan.groza@aut.upt.ro, Phone +40-256-403-242

change the environment in which we live by deploying cutting-edge infrastructure that would have been impossible to build on human power alone. More recently, the degree of autonomy of vehicles drastically improved. This happened not only by the introduction of various driver assistance technologies, such as automatic cruise control and autonomous emergency braking systems, but also with basic self-driving capabilities that are going to be extended until fully autonomous vehicles will travel the roads. Being such an important asset and now having such an enormous potential for being controlled by malicious pieces of software, it is no surprise that vehicles become a potential cybersecurity target.

Fortunately, so far, attacks on vehicles have been only demonstrative in nature, such as the experimental analysis provided the research in [10], a now famous attack on a Jeep car [22], and more recently some remote attacks on TESLA cars [26]. As security become manifest, the stakeholders had to react with standards and regulations that facilitate the deployment of security countermeasures and the proper incident response mechanisms. The array of standards ranges from AUTOSAR requirements on cryptography layers and secure on-board communication [5] to the more recently released requirements for intrusion detection systems on Electronic Control Units (ECU) [4]. In parallel to these, standards for security evaluation and for the assessment of security incidents were released. These include the more recent ISO 21434 [1] containing the cybersecurity guidelines which we will follow in this work.

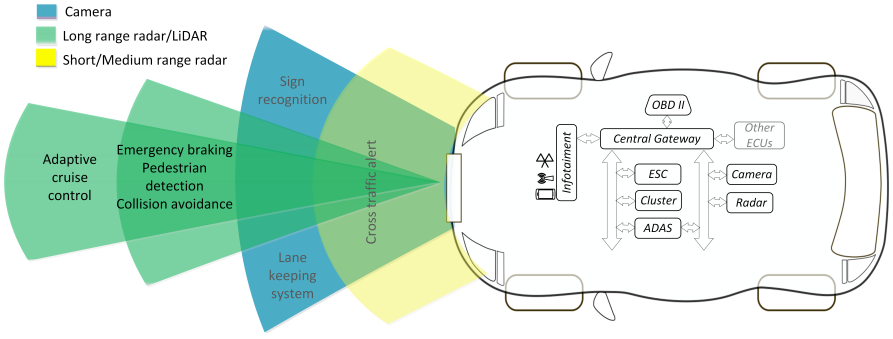


Fig. 1 Car equipped with camera and radar sensors for driver assistance or partial autonomous driving

The introduction of these standards provides enormous help for an industry which now produces almost 80 million vehicles each year. But the problems are far from being solved since specifications inside standards do not provide the exact procedures and security mechanisms to mitigate the attacks which are up to the manufacturer. And more, no security mechanism is perfect and manufacturers have to imagine clever ways to mitigate the attacks. For this reason, we will focus on a secure-by-design Automatic Emergency Breaking (AEB) system, a system which is intended

to trigger the brakes in order to avoid collisions with another vehicle or pedestrian. A car equipped with an AEB system and various other sensors is suggested in Figure 1. Various long or short-distance radars and cameras report data to an Advanced Driver-Assistance Systems (ADAS) ECU which decides to request braking to the Electronic Stability Control (ESC) ECU. More details about this architecture will be added in a forthcoming section. To facilitate a security analysis according to ISO 21434 [1] we will need to proceed to a more in-depth evaluation at the control system level, clarifying the security mechanisms that should be put in place as well as the effects of various types of attacks.

The exposition in this chapter is structured as follows. We begin by providing some background for the reader that is unfamiliar with the AEB system and ISO 21434 in Section 2. Then we proceed to an in-depth analysis on adversary actions and their impact on the AEB system in Section 3. This is the most demanding section of our work and is extremely important since it allows us to set room for the exact security specifications for such systems. In Section 4 we proceed to an analysis following the ISO 21434 activities leading to specific security goals. Finally, Section 5 holds the conclusion of our work.

2 Background

In this section we set a brief background on the AEB system that will serve to us as a case study and on ISO 21434 which provides the guidelines for a security-aware design.

2.1 The AEB system in a nutshell

The Automatic Emergency Braking (AEB) is one of the main ADAS functionalities designed to detect slow or stopped vehicles and pedestrians ahead and to trigger the brakes immediately. It is thus a system that can save the lives of passengers and of the traffic participants, pedestrians in particular. Other ADAS functionalities include the Adaptive Cruise Control (ACC) which is present in many older vehicles as well and the more recent Blind Spot Monitoring (BSM), Forward Collision Warning (FCW), Lane Departure Warning (LDW) and Lane Keeping Assist (LKA). As the names suggests, these systems ensure that the driver is signalled for the presence of objects on the sides (BSM), the approach toward a stationary object on the front (FCW) or the departure from the lane (LDW), eventually helping the driver by keeping the car to follow the lane (LKA). To provide a crisper case study, we will focus on the AEB system alone.

The scope of the AEB module is to prevent the accidents or to minimize the injuries resulting from such accidents by reducing the vehicle speed automatically when an obstacle, e.g., bicycle, pedestrian or sudden braking of the lead vehicle, is

detected with the help of the long-range radar and front camera. The AEB system has more than a decade of use, Volvo first introduced the system in 2009. Since 2014, the European New Car Assessment Program (Euro NCAP)[1] introduced specific evaluations for the autonomous braking in the AEB City and AEB Interurban tests for low speed and high speed scenarios. The AEB feature is available in the majority of the recently released cars thus becoming an ubiquitous functionality.

The AEB module is a safety component and is part of the Forward Collision Avoidance (FCA) system. In Figure 2 we give an overview of the AEB system function suggesting one vehicle that approaches a pedestrian. When the front camera and the front long-range radar detect the obstacle, an acoustic signal is activated and a visible warning light for the driver is displayed on the cluster. Afterwards, three braking stages follow: a first stage of slow pre-braking (partial braking), then a second stage of intensive pre-braking (partial braking) and finally the full braking stage. This image should be sufficient to understand the functionality of the AEB, the concrete system model will be detailed later.

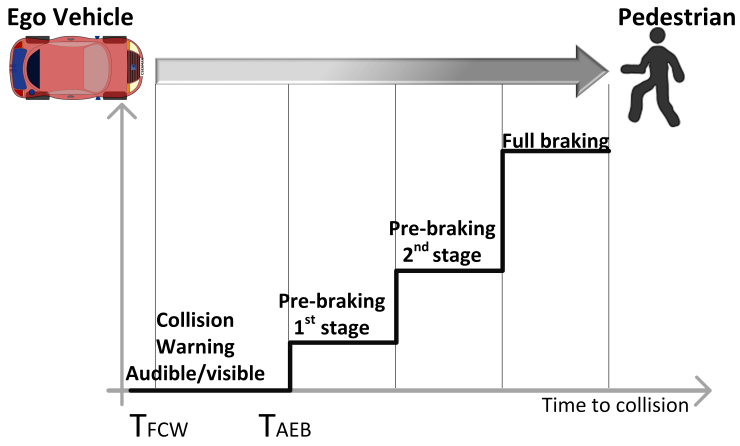


Fig. 2 Overview of the AEB system signalling and actions

2.2 Overview of ISO 21434 activities

The automotive industry heavily relies on the V model for the development cycle of in-vehicle components. There are good reasons behind this choice, the most relevant being the rigorous interaction between the design and testing stages. It is

[1] https://www.euroncap.com/en/vehicle-safety/safety-campaigns/2013-aeb-tests/

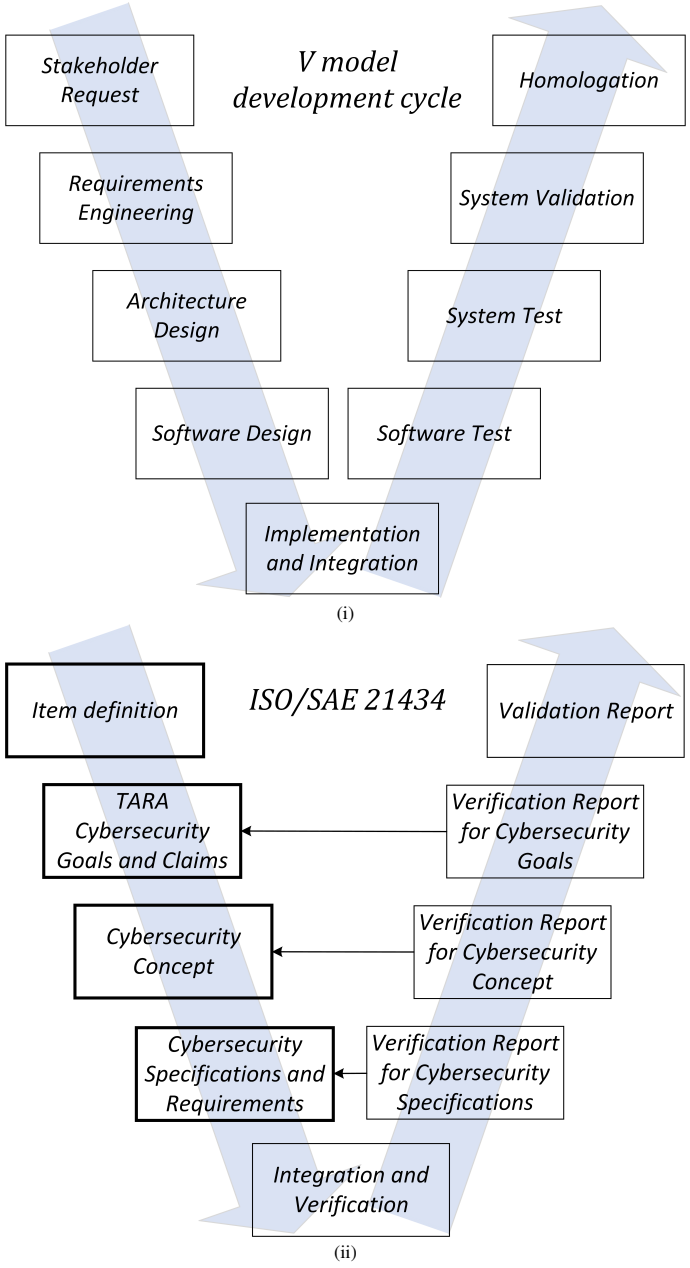


Fig. 3 Development cycle in the automotive industry as a V model (i) and cybersecurity related activities according to ISO 21434 expressed as a V-model (ii)

no coincidence that a similar view can be expressed for the cybersecurity-aware design of in-vehicle components. On the left side of Figure 3 we illustrate a generic development cycle in the automotive industry as a V model. It starts on the left branch with the stakeholder's requests which come from costumers and legislation, followed by requirement engineering, architecture design, software design, implementation and integration. On the right branch we have software testing, system testing, system validation and finally the homologation of the product. Similarly, on the right side of Figure 3 we depict as a V-model the cybersecurity related tasks according to ISO 21434 using some of the activities outlined in Annex A of the standard. They start on the left branch with the item definition, followed by a Threat Analysis and Risk Assessment (TARA) then the definition of the cybersecurity goals and claims, the cybersecurity concept, specification and requirements and finally the integration and verification. On the right branch there are verification reports for each of these steps culminating with a final validation. In the cybersecurity related V model from Figure 3 (ii) we highlight the first four steps, from item definition to cybersecurity specifications and requirements which are the subject of our analysis in this work. These four activities will be detailed for our AEB model in a forthcoming section.

Although it was published less than one year ago, it is worth mentioning that ISO 21434 has been already also used in several recent works. An earlier overview of ISO 21434 can be found in [19]. A whitepaper which places ISO 21434 in the context of other automotive and cybersecurity standards is also made available by BSI Group (British Standards Institution) [8]. A tool entitled ThreatGet which is compliant to the ISO 21434 is proposed by the authors in [31]. The authors exemplify the use of the tool that they design for an automotive gateway ECU. As underlined by the authors [31], the analysis which ISO 21434 facilitates is an asset driven security analysis which focuses on assets to determine the impact as well as the attack path - this requires specific treatment for each component. The authors in [28] present an ISO 21434 risk assessment methodology. The risk assessment they propose is based on an offline phase, which assess the damage scenarios and asset dependencies, etc., and on an online phase which assess the risk of a reported incident. Another attack surface assessment based on ISO 21434 is presented in [27]. Their analysis is mostly focused on the attack feasibility rating. This rating along with the impact rating can be used to determine the risks according to ISO 21434. The analysis that we perform here on the AEB system is in-line with the previously mentioned works as we follow the same specifications from ISO 21434. What differs is the component on which we focus and the in-depth adversary model and protection mechanisms at the control system level which are not present in the previously mentioned works.

It is also worth mentioning that there are several other works concerned with security assessments for automotive components, which were published well before the release of the ISO 21434. Maybe the earliest is [15] which tries to drive security requirements from various threat scenarios, including ECU corruption and spoofed CAN messages, etc. An analysis centered on the use of the Body Control Module (BCM) as the critical gateway component is done in [12]. The work in [14] performs a similar risk analysis but it is centered on vehicle instrument clusters which can be corrupted to mislead the driver and thus cause accidents. Another risk and counter-

measure analysis was done in [6]. The authors in [29] discuss a risk assessment and cybersecurity analysis based on ISO/IEC 27001. In [34] another cybersecurity risk assessment is discussed which accounts for several types of attacks on the CAN bus.

3 In-depth analysis of adversarial actions on AEB control systems

Existing research works that apply the ISO 21434 security standard, generally take a straight forward way in classifying adversary actions and their impact based on generic assumptions regarding the attack of a component. Here we will proceed to a more in-depth analysis that accounts for exact adversarial manipulations of CAN frames and we try to determine the exact impact that these actions will have on safety. This analysis is needed in order to accurately assess the risks and understand the countermeasures.

3.1 Detailed AEB system model

In order to simulate the vehicle reaction in case of distinct attacks on the CAN bus, we will use an existing Simulink model for Autonomous Emergency Braking with Sensor Fusion from MathWorks[2] and add adversarial behaviour to the model. Other works have also used Simulink models to test and validate attacks from the CAN bus. For example, the authors in [11] use a Simulink model of the Anti-lock Braking System (ABS) developed by Mathworks to test a multilevel monitor for the isolation and detection of attacks over the senors and CAN bus for a Cyber Physical System (CPS). In [18] an ACC model from Simulink is used to validate a method for the detection and mitigation of spoofing attacks on the radars which are used by the ACC system. The authors check the integrity of the radar sensor data based on a spatio-temporal challenge-response (STCR) which transmits signals in random directions and identifies then excludes signals reflected from untrustworthy directions.

In Figure 4 we depict the model of the AEB and the placement of six potential adversaries denoted as A1–A6. The Simulink AEB model contains two functional parts, the AEB functionality and the vehicle and environment component. The AEB functionality is also split in two ECUs: Camera/Radar ECU and AEB/ADAS ECU. The Camera ECU acquires data from the hardwired components, i.e., camera and radar, based on which it derives the information about obstacles and computes the relative distance and velocity which are transmitted to the ADAS ECU using the Private CAN bus communication (Pr-CAN). The ADAS ECU, in addition to this information received from the Camera ECU, also receives the longitudinal velocity from the ESC ECU using the Chassis CAN bus communication (C-CAN). The ADAS ECU implements an AEB controller which computes the deceleration request needed

[2] https://nl.mathworks.com/help/driving/ug/autonomous-emergency-braking-with-sensor-fusion.html

to stop the vehicle in order to avoid the collision with the obstacle, i.e., the pre-braking stages which is the AEB status. Additionally the ADAS ECU implements a speed controller (designed as a PID controller) to determine the required acceleration in order to maintain the ego velocity setpoint, i.e., the throttle. Finally, based on the AEB status and the internally computed acceleration, the ADAS ECU computes the throttle position. The deceleration request and the throttle position are transmitted over the C-CAN to the ESC ECU which controls the vehicle based on the requested commands. Additionally, in our model, the ESC ECU computes the vehicle position and trajectory represented as: XY position, XY velocity, yaw rate, yaw angle and longitudinal velocity. For simplicity, in the Simulink scenario, the curvature of the road is set as a constant. As stated, there are six adversary positions in Figure 4, representing the six signals which can be attacked in the model.

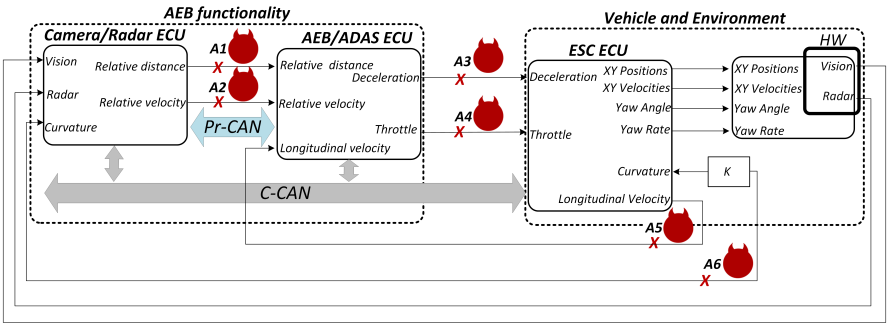


Fig. 4 The AEB model with the attach surfaces

3.2 Adversary model and attack strategies

For an accurate description of the attacks at the control system level, we need to formalize the adversarial actions. The adversary actions consist in manipulating a specific signal and we will use $y_{\blacklozenge}(k)$ to denote a signal at step k which is a positive integer, i.e., $k \in Z_N^+$, and $\blacklozenge$ is a placeholder to denote the six possible adversaries A1–A6 on six possible signals: deceleration (i.e., the braking stages), acceleration, ego velocity, relative distance, relative velocity and curvature (each corresponding to the 6 location points in Figure 4), i.e., $\blacklozenge \in \{\mathsf{brake}, \mathsf{throttle}, \mathsf{vego}, \mathsf{rdist}, \mathsf{rvel}, \mathsf{curvature}\}$. Then the value of $y_{\blacklozenge}(k)$ at each step will be either the legitimate signal $\overline{y}_{\blacklozenge}(k)$ or a signal originating from the adversary $\widetilde{y}_{\blacklozenge}(k)$. For example, $\overline{y}_{\mathsf{vego}}(k)$ will be the legitimate signal for ego velocity and $\widetilde{y}_{\mathsf{vego}}(k)$ is the adversarial signal corresponding for the ego velocity.

Most, if not all, of the existing works focusing on attacks and intrusion detection for CAN buses consider three types of attacks: replay, Denial of Service (DoS) and fuzzing attacks. These attacks can be easily formalized as follows:

1. *replay attacks* - by this attack, CAN frames are re-transmitted, possibly with a random delay, containing previously recorded signals, i.e., $\widetilde{y}_{\blacklozenge}(k) \leftarrow \overline{y}_{\blacklozenge}(i), i < k$
2. *fuzzing attacks* - are a modification attack in which random values are injected in the datafield of CAN frames, essentially meaning that the attack signal becomes a random value, i.e., $\widetilde{y}_{\blacklozenge}(k) \leftarrow$ rand,
3. *DoS attacks* - prohibit CAN frames from being transmitted on the bus and are specifically difficult to address since an adversary can always write high priority frames on the bus or even destroy legitimate frames with error flags or by distortions of the data-field that deem them unusable, which means that the signals are effectively lost, i.e., $\widetilde{y}_{\blacklozenge}(k) \leftarrow \perp$.

For all of the previous attacks, in the later model where we evaluate them, we assume a probability of occurrence, simply denoted as p, which is the probability of an adversarial signal (or CAN frame) to replace a legitimate one. Assigning a probability to the event of an attack is in line with the practical side of the problem since adversaries usually insert manipulated frames that compete with legitimate frames on the bus. This also responds to the situation in which an intrusion detection system is in place on the controller and only some of the adversary frames will go undetected and accepted as legitimate.

Still, these three types of attacks described above are insufficient for giving a complete image over both the attacker and defense capabilities. Notably, an intrusion detection system may be into place and arithmetic attacks which take advantage of the system model and inject specific values that are expected to cause a particular behavior of the car may be an option. Such a scenario has received little or no attention at all in the research literature related to car security. A reason for which will carefully examine three flavours of stealthy attacks which were also pointed out in well known control system security paper [9]. These include three flavours of stealthy attacks: surge attacks, bias attacks and geometric attacks [9].

For this reason, let us consider that an intrusion detection system may be in place. The problem addressed by an intrusion detection system is thus to distinguish between the two values $\overline{y}_{\blacklozenge}(k)$ and $\widetilde{y}_{\blacklozenge}(k), \forall k \in Z_N^+$. Since no intrusion detection system is perfect, a small false negative rate exists, i.e., some of the adversarial frames may go undetected. To introduce more specialized attacks and countermeasures, we may consider that a simple change detection algorithm stays at the core of the intrusion detection method implemented on the ECU. Such an algorithm may account for statistical distances between value and various range checks. The work in [9] dedicated to control systems security, suggests the use of cumulative sums (CUSUM) statistics over the reported value and some predicted value for the same signal. This methodology is indeed well suited for our scenario since we can infer the value of one signal from another signals available in the car. For example, using the relative velocity reported by a radar and the velocity of the car, we can compute the distance to the object and compare it with the reported one, etc. Generally speaking,

having the predicted value of the signal $y'_{\blacklozenge}(k)$ and a bias b we can use the following recurrent sum to detect an attack [9]:

$$S_{\blacklozenge}(k) = \max\{0, S_{\blacklozenge}(k-1) + |y_{\blacklozenge}(k) - y'_{\blacklozenge}(k)| - b\}, S(0) = 0$$

Whenever the error between the predicted value and the reported value is greater than the bias b, the error is added and, when the sum reaches a signalling threshold τ, the signal will be deemed as adversarial. That is, if $S_{\blacklozenge}(k) > \tau$ we consider $y_{\blacklozenge}(k)$ to be an attack signal and if $S_{\blacklozenge}(k) \leq \tau$ we consider $y_{\blacklozenge}(k)$ to be legitimate. Having this change detection procedure in mind, an adversary can mount the following three stealthy attacks that are described in [9]:

1. *surge attacks* - are the modification attacks in which the value of the signal is set to the maximum value (or minimum value) such that it will inflict the maximum damage on the system; to remain stealthy and go undetected by the cumulative summing, the attack value at step $k+1$ will be $y_{\blacklozenge,\max}$ only if the corresponding sum at the next step $S_{\blacklozenge}(k+1) \leq \tau$ while otherwise the attack signal will stay at $y'_{\blacklozenge}(k) + |\tau + b - S_{\blacklozenge}(k)|$ (note that in this way $|\tau + b - S_{\blacklozenge}(k)|$ is the maximum value that can be added to the legitimate signal such that an intrusion will not be detected).
2. *bias attacks* - are the modification attacks in which a small constant $c = \tau/n + b$ is added at each step to the attacked signal, i.e., $\widetilde{y}_{\blacklozenge}(k) \leftarrow y_{\blacklozenge}(k) + \tau/n + b$, ensuring that the attack remains undetected for n steps (this happens so since the threshold is divided over the n steps of the attack),
3. *geometric attacks* - are the modification attacks in which a small drift is added to the attacked signal in the beginning and the drift becomes increasingly larger in the next steps using a geometric expansion, i.e., $\widetilde{y}_{\blacklozenge}(k) \leftarrow y_{\blacklozenge}(k) + \beta\alpha^{n-k}$ where α is fixed and $\beta = \frac{(\tau+nb)(\alpha^{-1}-1)}{1-\alpha^n}$.

To sum up, in the light of these attack strategies, we are concerned with assessing the impact of two kinds of adversarial actions: those attacks that will go undetected due to the non-zero false negative rate of the in-vehicle IDS and the stealthy attacks in which adversary actions deviate by a small margin from the predicted values, thus remaining undetected. We discuss the impact of the attacks in what follows.

3.3 Attack evaluation on the AEB model

One of the test scenarios from the Euro NCAP car safety performance assessment programme [3] is the Car-to-Pedestrian Nearside Child test dedicated to the AEB functionality. In Figure 5 we depict this scenario. Two vehicles are stationary on the right side of the road and there is a pedestrian nearby, crossing the road at one meter from the cars. The ego vehicle is travelling on the left lane of the road, the view of the pedestrian crossing is obstructed for the driver. The AEB system has to activate the automatic braking in order for the car to stop and avoid collision with the pedestrian.

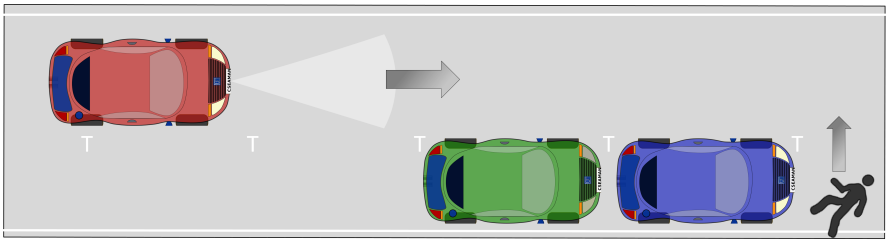


Fig. 5 Overview of the AEB scenario

This scenario is used in the Simulink model as well and we will analyze the adversarial impact on it. We consider attack points A1–A5 and leave A6 outside the discussion since A6 represents the curvature of the road and attacking this value will lead the vehicle outside the lane, not causing a collision with the pedestrian in front which is our attack target. An attack on the curvature is relevant, but will not fit our specific use case. More, the countermeasure which we later propose will hold for manipulating data related to A6 as well.

Signal interpretation. To clarify the impact of the attacks, in Figures 6, 7 and 8 we illustrate the signals starting from a clear scenario without adversarial interventions and then add several types of attacks to the signals. We show the plot markers for each signal in Figure 6 which corresponds to the case when there is no adversarial intervention. We use two types of plots: (i) FCW/AEB status plots which show the status of the collision warning and deceleration/acceleration stages and (ii) velocity/distance plots which show the velocities and the distances toward the object in front. Note that each plot has distinct axes on the left and right side. For the FCW/AEB status, we plot the status signals on the left axis (marked with black) and the accelerations measured in m/s^2 on the right axis (marked with gray). The status signal from the left axis includes the signal which indicates the activation of the FCW with solid line, AEB status (braking stages) with dashed line and the signal which indicates that the ego car was stopped which is marked with a dashed-dotted line. On the right axis, we plot the deceleration with dotted line and the acceleration with a dotted line marked by circles. For the velocity/distance plots, we again plot two axes, one for the velocity and another one for the distance. The signals plotted on the left axis (marked with black) are the following: the preset velocity for the vehicle marked with dashed-dotted line, the ego velocity marked with solid line and the relative velocity marked with dashed line. The signals plotted on the right axis (marked with gray) are the following: the relative distance marked with dotted line and the headway marked with dotted-circle line.

Attack-free scenario. A few words on the plots for the attack-free scenario may be helpful. In Figure 6 we illustrate the signals without the adversarial interventions. As the relative distance between the ego car and the obstacle becomes lower than 15 meters (velocity/distance plot), the FCW system becomes active (FCW/AEB status) and the AEB begins the braking stage for the car. As expected, the AEB status follows

the three pre-braking stages until the car is stopped. The acceleration is decreasing during braking, thus the ego velocity is decreasing from the preset velocity until the car eventually stops. The dotted line marked with circles shows the headway which is the difference between the relative distance and the length of the car (the headway is about 2 meters when the cars stops).

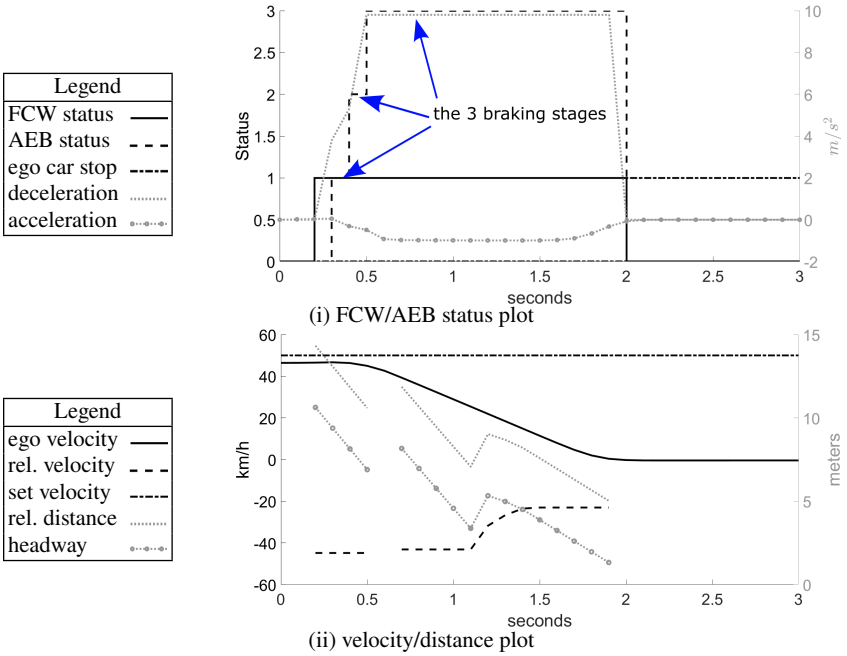
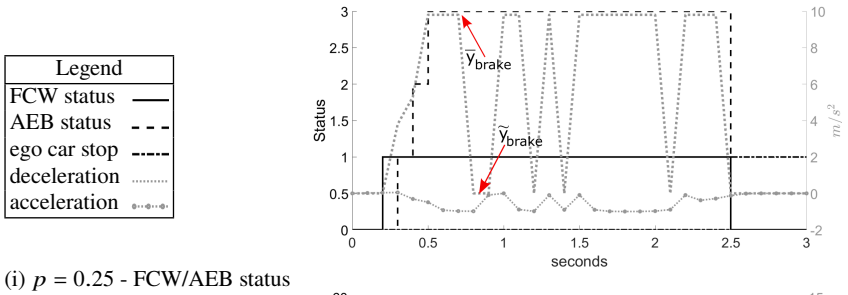


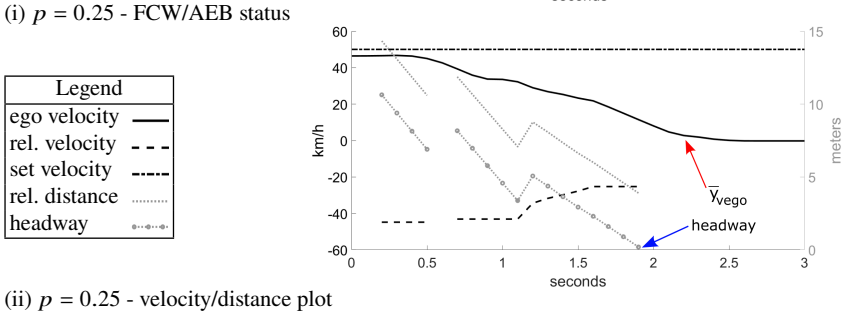
Fig. 6 Signals without adversarial intervention

Replay attacks. In Figure 7 we depict the AEB signals under a replay attack on the braking (deceleration) signal and the adversarial signal $\widetilde{y}_{\text{brake}}(k)$ corresponding to the deceleration. To inflict maximum damage, the replayed valued for the deceleration is the minimum value known to the intruder, i.e., $\widetilde{y}_{\text{brake}}(k) = 0$. We will use the following attack probabilities: $p = 0.25$, $p = 0.5$ and $p = 0.75$. In the FCW/AEB status plots from Figure 7 it can be easily observed that the adversarial signal $\widetilde{y}_{\text{brake}}(k)$ corresponding to deceleration (gray dotted line) is set to zero more often when the attack probability is increasing, which means that the adversary deactivates the brake request more often. In the velocity/distance plots from the Figure 7 we depict the response of the car to the attack. The ego velocity (black solid line) in Figure 7 (ii) is 0 km/h when the headway is 0.19 meters which means that there is no impact at an attack probability of $p = 0.25$. In Figure 7 (iv), the ego velocity is 25.53 km/h when the headway is 0 meters which means that at an attack probability of $p = 0.5$ the impact will take place. In Figure 7 (vi), the ego velocity is 42.08 km/h when the headway is 0 meters which means that at an attack probability of

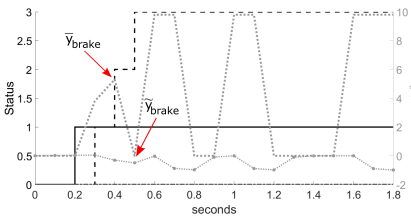
$p = 0.75$ again there will be an impact at a considerable speed. As expected, the impact severity of the attack is increasing with the attack probability.



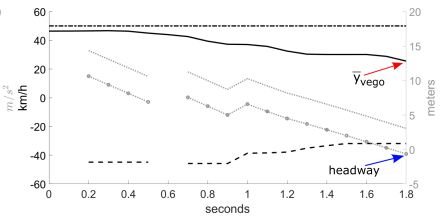
(i) $p = 0.25$ - FCW/AEB status



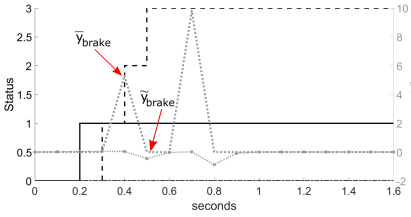
(ii) $p = 0.25$ - velocity/distance plot



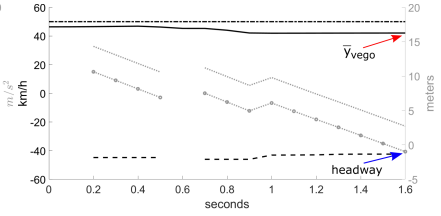
(iii) $p = 0.5$ - FCW/AEB status



(iv) $p = 0.5$ - velocity/distance plot



(v) $p = 0.75$ - FCW/AEB status



(vi) $p = 0.75$ - velocity/distance plot

Fig. 7 Signals after a replay attack on deceleration y_{brake} with distinct attack probabilities: (i) $p = 0.25$ - FCW/AEB status, (ii) $p = 0.25$ - velocity/distance plot (iii) $p = 0.5$ - FCW/AEB status, (iv) $p = 0.5$ - velocity/distance plot, (v) $p = 0.75$ - FCW/AEB status and (vi) $p = 0.75$ - velocity/distance plot

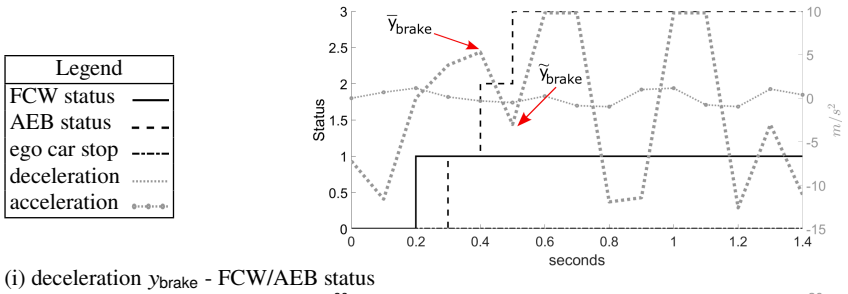
Fuzzing attacks. In Figure 8 we illustrate the signals when fuzzing attacks take place, with an attack probability of $p = 0.5$, on three attack surfaces A1, A3 and A4. In Figures 8 (i) and (ii) we depict the impact for a fuzzing attack on the deceleration signal. The adversarial signal $\widetilde{y}_{\mathsf{brake}}(k)$ corresponding to deceleration (gray dotted line) has random values which reduce the brake intensity or even deactivate the brake request in order to produce impact. The ego velocity (black solid line) is 50.54 km/h when the headway is 0 meters which means that when the fuzzing attack on deceleration takes place with probability of $p = 0.5$ it induces an impact at considerable speed. In Figures 8 (iii) and (iv) we show the fuzzing attack on acceleration. The adversarial signals $\widetilde{y}_{\mathsf{throttle}}(k)$ do not significantly impact the functionality of the AEB system. The car is stopped in time, in some cases only the FCW is activated after the car is stopped. In Figures 8 (v) and (vi) we depict the impact of a fuzzing attack on the relative distance signal. The adversarial signal $\widetilde{y}_{\mathsf{rdist}}(k)$ corresponding to the relative distance (gray dotted line) produces a delay in the activation of the AEB system which causes an impact at a velocity of 45.61 km/h.

Table 1 AEB results: collision velocity and distance to target in case of replay, fuzzing and DoS attacks at various success rates

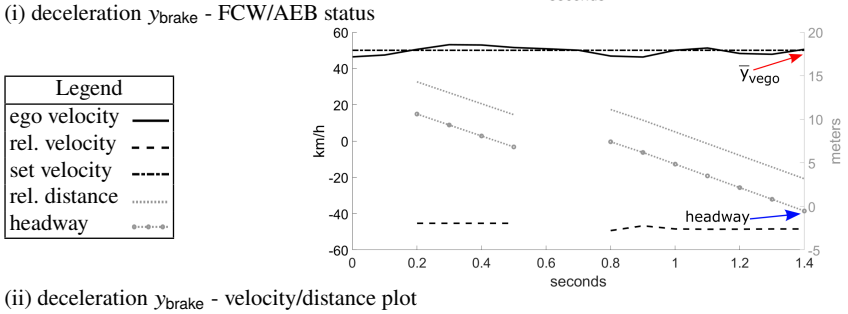
Attack	**Signal**	$p = 0.25$ (0.2 for DoS)		$p = 0.5$		$p = 0.75$ (0.7 for DoS)	
		Collision velocity[km/h]	**Distance to target[m]**	**Collision velocity[km/h]**	**Distance to target[m]**	**Collision velocity[km/h]**	**Distance to target[m]**
Replay	A3: Deceleration	no coll.	0.19	25.53	0	42.08	0
Fuzzing	A3: Deceleration	43.99	0	50.54	0	57.27	0
	A4: Throttle	no coll.	2.32	no coll.	2.37	no coll.	2.26
	A1: Relative distance	33.89	0	45.61	0	45.61	0
	A2: Relative velocity	no coll.	1.32	no coll.	1.32	no coll.	1.32
	A5: Long. velocity	no coll.	1.46	no coll.	1.46	no coll.	1.90
DoS	A3: Deceleration	no coll.	1.32	9.01	0	32.33	0

DoS attacks. In case of DoS attacks we chose to apply the attack again only on deceleration signal. Since our simulation is running with a step of 0.1s we can simulate only the attack probabilities which are multiple of 0.1, i.e., $p = 0.2$, $p = 0.5$ and $p = 0.7$. For brevity, we omit the plots for this attack scenario and we refer the reader to the results in Table 1. The DoS attack on deceleration does not cause an impact for an attack probability $p = 0.2$, but it produce impact with a collision velocity of 9.01km/h and 32.33km/h in case of attack probabilities $p = 0.5$ and $p = 0.75$ respectively. We use "no coll." to denote that no collision took place.

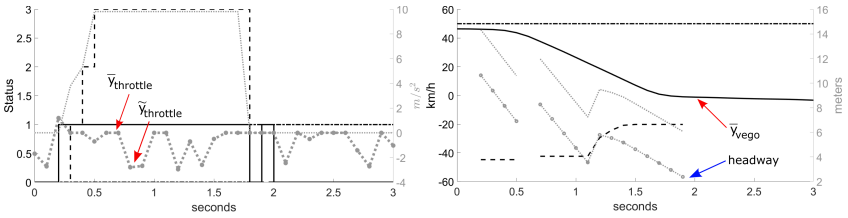
Also, in Table 1 we summarize as numerical data the collision velocity and distance to target in case of replay and fuzzing at various attack success rate, i.e., $p = 25$, $p = 0.5$ and $p = 0.75$. In case of the replay attack we apply the attack only on the deceleration signal since it has the most significant effect. The attacks are causing a collision at $p = 0.5$ and $p = 0.75$ and the collision velocity is increasing with the attack probability. In case of fuzzing attacks, as can be already observed in the previous figures, no collision happens when the adversarial signals are $\widetilde{y}_{\mathsf{throttle}}(k)$,



(i) deceleration y_{brake} - FCW/AEB status

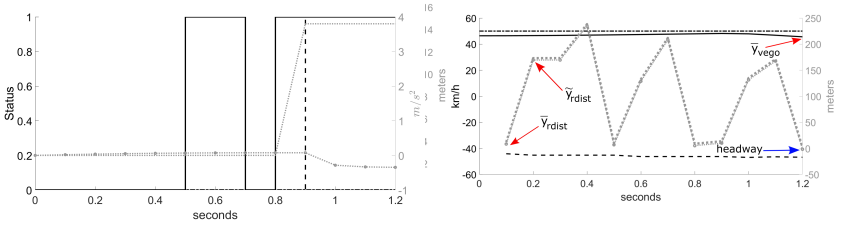


(ii) deceleration y_{brake} - velocity/distance plot



(iii) acceleration $y_{throttle}$ - FCW/AEB status

(iv) acceleration $y_{throttle}$ - velocity/distance plot



(v) relative distance y_{rdist} - FCW/AEB status

(vi) relative distance y_{rdist} - velocity/distance plot

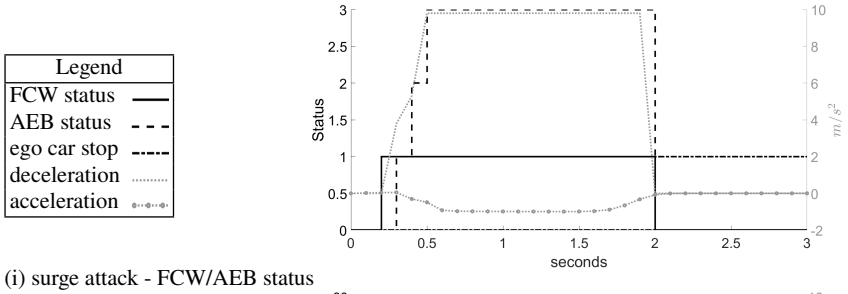
Fig. 8 Signals under fuzzing attack with attack probability of $p = 0.5$ on: (i) deceleration y_{brake} - FCW/AEB status, (ii) deceleration y_{brake} - velocity/distance plot, (iii) acceleration $y_{throttle}$ - FCW/AEB status, (iv) acceleration $y_{throttle}$ - velocity/distance plot, (v) relative distance y_{rdist} - FCW/AEB status and (vi) relative distance y_{rdist} - velocity/distance plot

$\widetilde{y}_{\mathsf{rvel}}(k)$ and $\widetilde{y}_{\mathsf{vego}}(k)$ corresponding to the throttle, relative distance and longitudinal velocity. On the other hand, when the fuzzing attacks are applied on deceleration and relative distance, the collision takes place at all attack probabilities with an impact velocity which increases with the attack success rate from 43.99km/h to 57.27km/h in case of the deceleration and from 33.89km/h to 45.61km/h in case of the relative distance.

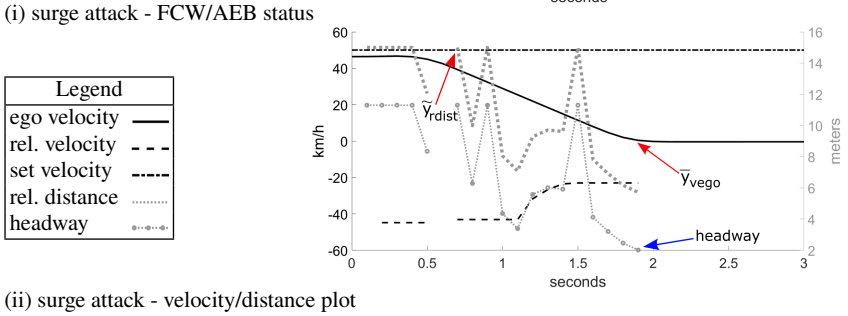
3.4 Impact of stealthy attacks

We now discuss the impact of the three types of stealthy attacks: surge attacks, bias attacks and geometric attacks. The work in [9] evaluated the impact of these attacks on a control system for a chemical reactor process and we will now evaluate it on our AEB system. This type of attacks assume that a change detection mechanism is in place. Distinct from the work in [9], we do not use a state predictor to infer the next value but we can infer the value of some of the parameters from the others and use this in the cumulative sum. Namely, we estimate the relative distance from the relative velocity along with the longitudinal velocity and we also estimate the deceleration as the derivative of the longitudinal velocity. The estimated values of the signals are used in place of $y'_{\blacklozenge}(k)$ when computing the cumulative sum $S_{\blacklozenge}(k)$. The rest is similar in the change detection mechanism and in the computation of the attack values. In what follow we will demonstrate the impact of stealthy attacks on the relative distance and deceleration. The attacks on the relative distance will have little effects and will not cause a collision, while the attacks on deceleration will lead to collisions at significant speed.

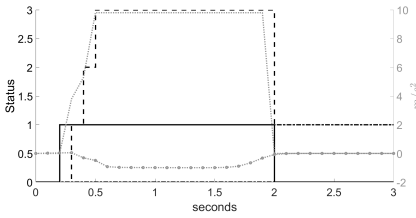
Stealthy attacks on relative distance. In Figure 9 we illustrate the signals when a stealthy attack on relative distance takes place, i.e., the adversarial signal is $\widetilde{y}_{\mathsf{rdist}}(k)$. In the FCW/AEB status plots from Figure 9 it can be seen that the AEB functionality is not influenced by the stealthy attacks on relative distance as the car stops in time to avoid the collision. This is because the stealthy attack cannot take advantage of the larger random values of the previously demonstrated fuzzy attack (this will make the attack detectable). In the velocity/distance plots of Figure 9 we show the adversarial signal $\widetilde{y}_{\mathsf{rdist}}(k)$ corresponding to the relative distance and the headway which is also influenced by the attack. In Figure (ii), corresponding to surge attacks, the adversarial signal $\widetilde{y}_{\mathsf{rdist}}(k)$ sets the relative distance to 15 meters (maximum value of the relative distance in normal conditions) on several points. But still, no impact occurs. In Figure (iv), corresponding to a bias attack, the adversarial signal $\widetilde{y}_{\mathsf{rdist}}(k)$ smoothly distorts the relative distance signal placing it slightly below the real distance - still, there is no collision. In Figure (vi), corresponding to a geometric attack, the adversarial signal $\widetilde{y}_{\mathsf{rdist}}(k)$ starts from the real signal value and progressively increases in time in order to maximize the damage at the end. But again, there is no collision. Thus, in our simulation, none of the stealthy attacks on the relative distance influenced the AEB functionality in such way as to cause an accident. The effects will become more serious when the deceleration is attacked in the same way.



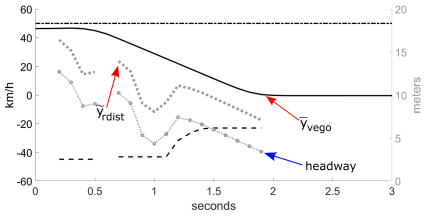
(i) surge attack - FCW/AEB status



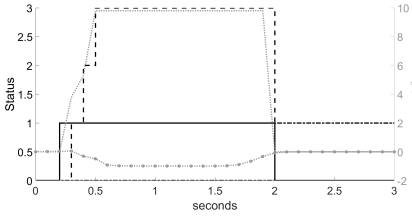
(ii) surge attack - velocity/distance plot



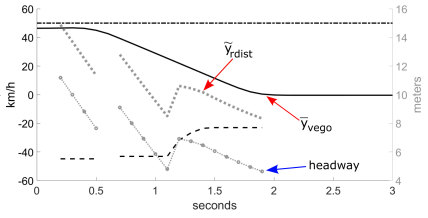
(iii) bias attack - FCW/AEB status



(iv) bias attack - velocity/distance plot



(v) geometric attack - FCW/AEB status

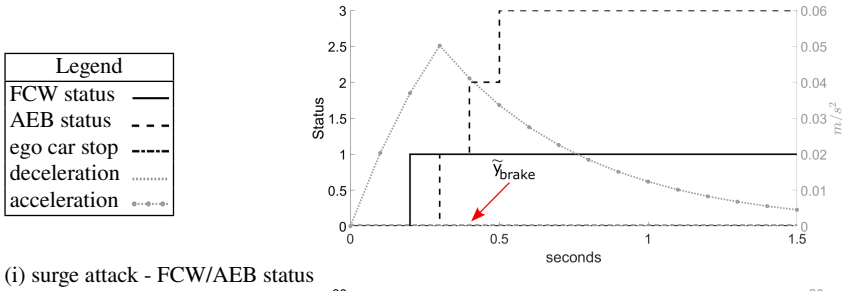


(vi) geometric attack - velocity/distance plot

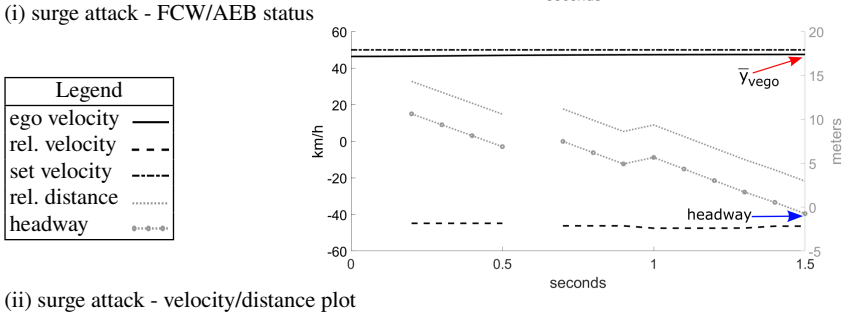
Fig. 9 Signals under stealthy attacks on relative distance y_{rdist}: (i) surge attack - FCW/AEB status, (ii) surge attack - velocity/distance plot (iii) bias attack - FCW/AEB status, (iv) bias attack - velocity/distance plot, (v) geometric attack - FCW/AEB status and (vi) geometric attack - velocity/distance plot

In terms of parameters for the stealthy attacks on relative distance, we did set the bias $b = 2$ because the maximum error between the signal computed by the ECU and the predicted signal is around 2 meters. The threshold was set equal with the bias, i.e., $\tau = 2$ and the number of steps was set to $n = 25$ for the bias and geometric attacks, because our simulation has 50 steps in case when no attack takes place and as such we will obtain the full attack during the first half of the simulation. For the geometric attack we set parameter $\alpha = 0.9$ to maximize the attack impact. Different parameters may yield distinct results.

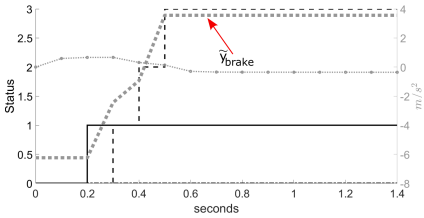
Stealthy attacks on deceleration. In Figure 10 we illustrate the signals when a stealthy attack on deceleration takes place, i.e., the adversarial signal is $\widetilde{y}_{\mathsf{brake}}(k)$. Now the attack will clearly result in an impact with the pedestrian. In Figures (i) and (ii) we depict the effect of a surge attack on the deceleration signal. In the FCW/AEB status plots of the figure, the adversarial signal $\widetilde{y}_{\mathsf{brake}}(k)$ is always $0m/s^2$ (gray dotted line) because in the attack implementation $y_{\min} = 0$, the acceleration is small (close to $0m/s^2$) as the vehicle speed is constant and in the velocity/distance plots it can be observed that the ego velocity remains near the preset velocity, i.e., no brake request comes from the AEB controller. Even if the AEB status is correctly shown in the FCW/AEB status plot, which follows the 3 braking stages, the deceleration request signal received by the ESC ECU is corrupt, requesting no deceleration, and thus the car continues to maintain the preset velocity. Note that the AEB status parameter is internal to the AEB controller and only the deceleration is communicated to the ESC ECU for the car to decelerate (this can be easily seen in the model from Figure 4). Therefore in the plot from Figure 10, while the AEB status is still set to 3 (full braking) inside the AEB controller, the deceleration value is subject to a stealthy attack and is much lower, misleading the ESC controller that the car should not brake and eventually leading to a collision. This attack causes an impact at an ego velocity of 47.51km/h (note that the headway is 0 meters) which means that the surge attack on deceleration creates an impact at considerable speed. In Figures (iii) and (iv) we depict the effect of the bias attack on the deceleration signal. In the FCW/AEB status plot, (iii), the adversarial signal $\widetilde{y}_{\mathsf{brake}}(k)$ (gray dotted line) is increasing until $4m/s^2$ are reached, but in normal conditions the deceleration should reach a much higher $10m/s^2$ (Figure 6). This leads to a much slower braking even if the AEB status request is set to full braking. In the velocity/distance plot (iv), the ego velocity is still near the preset velocity, reaching around 44.42 km/h at the time when the headway is 0 meters, i.e., when the collision occurs. This again means that the bias attack on deceleration produces an impact at considerable speed. Finally, in Figures (v) and (vi) we depict the effect of a geometric attack on the deceleration signal. In Figure (v), the adversarial signal $\widetilde{y}_{\mathsf{brake}}(k)$ (gray dotted line) is increasing until $8m/s^2$ are reached after which the geometric attack occurs and maximizes the damage as it abruptly decreases and the deceleration request gets near $0m/s^2$. In Figure (vi), the effect of this attack can be observed as the ego velocity is decreasing to 23.37km/h when the headway is 0 meters, i.e., the time of collision. This is a slightly lower impact velocity compared to the other two stealthy attacks. Still, all the three stealthy attacks on deceleration had caused an impact while they remained undetected by the change detection mechanism. We also note that in this case, another protection



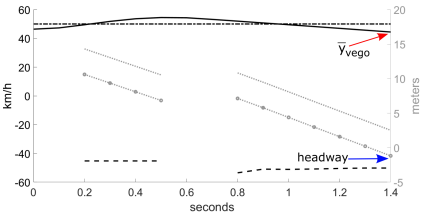
(i) surge attack - FCW/AEB status



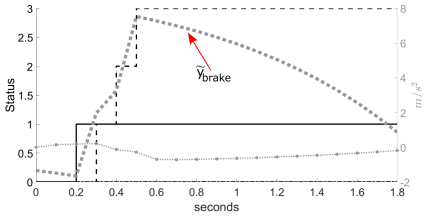
(ii) surge attack - velocity/distance plot



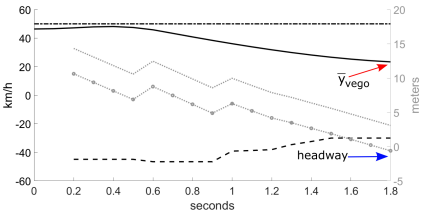
(iii) bias attack - FCW/AEB status



(iv) bias attack - velocity/distance plot



(v) geometric attack - FCW/AEB status



(vi) geometric attack - velocity/distance plot

Fig. 10 Signals under stealthy attacks on deceleration y_{brake}: (i) surge attack - FCW/AEB status, (ii) surge attack - velocity/distance plot (iii) bias attack - FCW/AEB status, (iv) bias attack - velocity/distance plot, (v) geometric attack - FCW/AEB status and (vi) geometric attack - velocity/distance plot

mechanism may be put in place: as the AEB controller orders the car to decelerate and the reported velocity does not decrease according to the expectations, the AEB controller may determine that an attack takes place. However, the ESC controller will not know which of the frames are the legitimate ones, i.e., lower or higher deceleration, and cannot act to correct the issue.

In terms of parameters for the stealthy attacks on deceleration, we set the bias $b = 6$ because the maximum error between the signal computed by the ECU and the predicted signal is around 6 m/s^2. The threshold was set equal with the bias, i.e., $\tau = 6$, the number of steps and parameter α were set again to $n = 25$ and $\alpha = 0.9$ respectively.

Table 2 summarizes in terms of numerical data the collision velocity and the distance to the target in case of stealthy attacks on deceleration and relative distance. The collision occurs in case of all the three attacks on deceleration signal, while no collision occurs when the stealthy attacks are applied to the relative distance as discussed previously. We use "no coll." to denote that no collision took place.

Table 2 AEB results: collision velocity and distance to target in case of stealthy attacks or deceleration and relative distance

Attack	Signal	Collision velocity[km/h]	Distance to target [m]
Surge		no coll.	2.01
Bias	A1: Relative distance	no coll.	3.40
Geometric		no coll.	4.63
Surge		47.52	0
Bias	A3: Deceleration	44.42	0
Geometric		23.37	0

4 Secure-by-design AEB in accordance to ISO 21434

Benefiting from the previous attack analysis, we will now follow the steps of ISO 213434 in order to point out specific security goals and the means to assure them for the AEB system.

4.1 Overview of the ISO 21434 cybersecurity design flow

For an accurate overview of the steps required by ISO 21434, we will first introduce an operational overview of the activities presented in the standard. These activities will be then detailed with respect to the AEB system that we use as an example. Figure 11 gives an overview of the activities presented in ISO 21434. These activities

are also exemplified in case of a headlight system which is used as a case study inside the standard.

The security related activities start from the definition of the item which is going to be secured, the boundaries of the system in which it is incorporated, its functions and a preliminary architecture. All these form the first step of the concept phase. A more tedious step follows which consists in the TARA (Threat Analysis and Risk Assessment). This step asks us to delve into more details regarding the identification of assets that need to be protected (either logical assets such as CAN frames, or physical assets as a specific sensor), rating the impact, identifying the threat, attack path, rating the attack feasibility, determining the risk level and the treatment option. The last step of the concept phase consists in the determination of the cybersecurity goals, claims and the introduction of the concept. After a correct understanding of the security goals, claims and concept are available, the specifications and requirements will be detailed in the product development phase. Next, the specifications are reviewed and then the product is integrated and verified, e.g., by the use of penetration testing tools, etc. Then the final validation report is completed. We address the concept phase according to ISO 21434 for the AEB system in what follows.

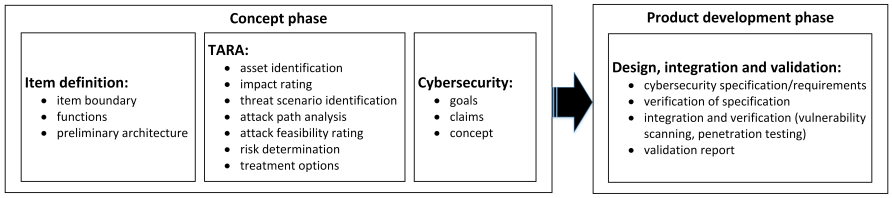


Fig. 11 Example of activities flow according to ISO 21434

4.2 From item definition to risk determination

Item definition. The first step of the concept phase is the item definition which includes the boundary, functions of the item and its preliminary architecture. The item boundary for the AEB system is presented in Figure 12. It includes the interfaces with internal and external items, forming as such the environment in which the AEB system resides. The function of the AEB, i.e., assisting the driver in avoiding collision with front objects, has already been clarified. The preliminary architecture of the AEB is encircled in the middle and it includes two CAN buses: the Chassis CAN (C-CAN) which is used for data exchange between the ADAS ECU, the ESC ECU and the cluster and the Private CAN bus (Pr-CAN) which is used for data exchange between the ADAS ECU, Camera and Radar.

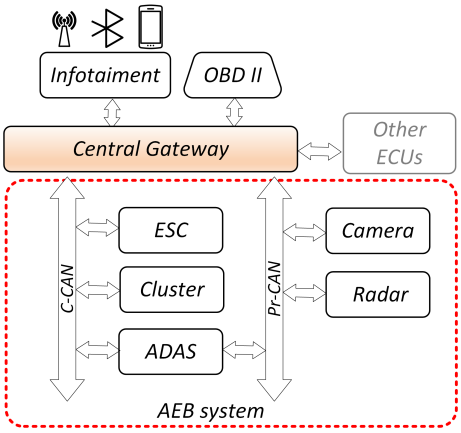


Fig. 12 In-vehicle network architecture and item boundary for the AEB system

Once the item is defined, a series of activities follow starting with the identification of the asset which needs to be protected up to the determination of the risk level. In Figure 13 we show these activities according to ISO 21434. For an easier understanding of these steps, we formulate one question for each step which summarizes its expected outcome. Next, we will address all the steps in this image for the AEB.

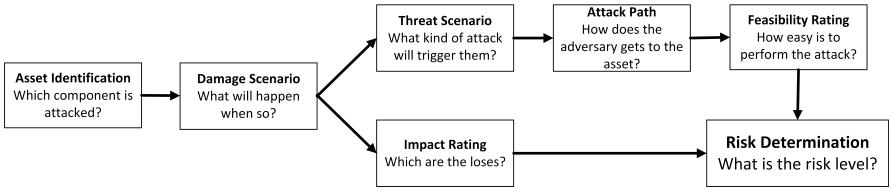


Fig. 13 Steps for risk determination according with to ISO 21434

Asset identification. The first step consists in identifying the assets that we are going to protect. These assets include logical objects such as the firmware or the CAN frames. Due to obvious space constraints for the current work, we will focus on CAN frames alone and consider that the rest of the components, such as the firmware and the hardware components are secured, e.g., by digitally signed software updates and the appropriate tamper resistant hardware such as TPMs (Truster Platform Module).

According to ISO 21434, the CAN frames as an asset, have to respond to the three classical security objectives: Confidentiality, Integrity and Availability (CIA). Confidentiality may not be a necessary requirement since there is no need to hide the content of the frames from an adversary (there are no privacy concerns). But integrity and availability are critical. The former has to ensure that the content has

not been modified or injected by an adversary, while the later must ensure that the corresponding frames are delivered on time by the ECU.

In Table 3 we detail the consequences when the two objectives, i.e., integrity (I) and availability (A), are not met for the frames which carry the signals from the AEB system. In the third column, the damage inflicted on the system is outlined. This includes the activation/deactivation of brakes, inability of the vehicle to maintain speed or estimate the time or distance toward the front object, etc. Generally, the attacks on integrity will mislead the car or the AEB system on the distance to the object or the time to collision, etc. The attacks on the availability of the signal will simply disable the corresponding functionality in the absence of required data.

Table 3 Asset identification and impact assignment:

Asset	Obj.	Damage scenario	Safety	Justification
Deceleration (CAN frame)	I	**D1.** Unexpected activation or deactivation of the brakes or jumping from one braking stage to another	severe	Can lead to accidents at high speed, see attack assessment from Sections 3.3 and 3.4
	A	**D2.** Unable to activate AEB function	moderate	The AEB can't activate the brakes and stop the car, driver may be warned of the lost functionality
Throttle (CAN frame)	I	**D3.** Unexpected self acceleration/ deceleration	severe	Unexpected self acceleration/deceleration may produce accidents at high speed
	A	**D4.** Car unable to self accelerate and maintain preset velocity	moderate	Unexpected loss of throttle signal would slow down the car but the driver should eventually notice this and compensate for the correct speed, visible/audible signal may also warn the driver for loss function
Relative Distance (CAN frame)	I	**D5.** AEB system is mislead on the correct distance to front obstacle, results in unexpected activation/deactivation of the AEB system	severe	Similar to the integrity (I) attack on throttle, may produce severe accidents
	A	**D6.** AEB system is unable to estimate the distance to the front object	moderate	Similar to the availability (A) attack on throttle, may still be noticeable for the driver that can compensate
Relative Velocity (CAN frame)	I	**D7.** AEB system is mislead on the correct time to collision, unexpected activation/deactivation of the AEB system	severe	Similar to the integrity (I) attack on throttle, may produce severe accidents
	A	**D8.** AEB system unable to estimate time to collision	moderate	Similar to the availability (A) attack on throttle, may still be noticeable for the driver that can compensate
Ego Velocity (CAN frame)	I	**D9.** AEB system is mislead on the correct acceleration request, results in self acceleration/deceleration	severe	Similar to the integrity (I) attack on throttle, may produce severe accidents
	A	**D10.** AEB system unable to compute the acceleration request, results in car unable to self accelerate and maintain the preset velocity	moderate	Similar to the availability (A) attack on throttle, may still be noticeable for the driver that can compensate

Impact rating. Impact rating is divided on four distinct chapters: safety (S), financial (F), operational (O) and privacy (P). A few justification on how we select these values in Table 3 may be needed. First, our table contains only the value for the safety impact since the financial, operational and privacy impact is identical as we argue at the end of this paragraph. The unexpected deactivation of automatic braking, in case of signal integrity which complies to the adversary manipulations

from the previous section, can have fatal consequences on the life of pedestrians, etc. For this reason, we consider attacks on integrity to have a severe impact due to the fatal injuries that may result from accidents. In case when the communication is lost, i.e., a DoS attack, the driver may still be warned by a visible or audible signal from the instrument cluster, showing that the functionality is not responding and as such he should increase his vigilance. A reason for which we consider that the loss of availability will have a more moderate safety impact. The same impact assessment holds for the rest of the parameters: throttle, relative distance, relative and longitudinal velocity. Not last, it is worth mentioning that based on our detailed analysis from Section 3.3 and 3.4, attacks on relative and longitudinal velocities will have more impact at higher speeds. The current version of ISO 21434, explicitly states that the financial impact refers to the costs of the road user. In most instances, car insurance companies cover these costs, although they may only cover the repair costs of the cars that are damaged by an inattentive driver and not the costs to repair their own car. We will consider that the financial costs should be moderate in general, although we cannot exclude that the financial impact may also run up to major, e.g., in case of impact at high velocities and the lack of the appropriate insurance. The operational impact is moderate since in case when the AEB functionality is lost, there is partial degradation of a vehicle function but the car is still fully controllable by the driver who is still able to brake. The privacy risks should be negligible.

Table 4 Risk determination for deceleration under the first two damage scenarios D1 and D2

Damage scenario	Threat Scenario	Attack Path	Feasibility Rating	Risk Value
D1. Unexpected activation or deactivation of the brakes or switching between braking stages	**T1**. Replay, **T2**. Fuzzing, **T3**. Stealthy attacks	OBD II connector	High	5
		Cellular interface	High	5
		Corrupted applications (3rd party)	Low	3
		USB port	Medium	4
		Malicious software (malware)	Low	3
		Software/hardware vulnerabilities from the development process	Low	3
		Unauthorized hardware added	Low	3
		Corrupted software update	Low	3
D2. Unable to activate AEB function	**T4**. DoS	OBD II connector	High	3
		Cellular interface	High	3
		Corrupted applications (3rd party)	Low	2
		USB port	Medium	2.5
		Malicious software (malware)	Low	2
		Software/hardware vulnerabilities from the development process	Low	2
		Unauthorized hardware added	Low	2
		Corrupted software update	Low	2

Threat identification. This step has been refined by the specific attacks embedded in our adversary model. The threat to availability (A) is posed by the DoS attacks. When it comes to integrity (I) our model accounts for different kinds of manipulations, i.e., replay, fuzzing, surge, bias and geometric attacks. Separating these threats is relevant, because they can be addressed in different ways as we will discuss later when introducing the cybersecurity requirements, e.g., some of the attacks can be

addressed by simple change detection mechanism while others require cryptographic authentication, etc.

Attack path analysis. For the attack paths we considered the regulations concerning the approval of vehicles with regard to cybersecurity and cybersecurity management systems [2] proposed by the United Nations Economic Commission for Europe (UNECE). These regulations were recently investigated by the authors in [7]. Due to page limitations, we only use here the most significant attack surfaces, e.g., OBD (On-Board Diagnostics) connector, cellular interface, USB ports, etc.

Attack feasibility rating. According to the specifications in the Annex G of the standard [1], attack feasibility is a mixture between 5 components: required time, expertise, knowledge of the component, window of opportunity and equipment. The aggregate attack potential resulting from the scores of these 5 components ranges from very low to high. For simplicity we will not detail the score based on each of the previous 5 components, but provide some arguments for the aggregate rating that we present in Table 4. The OBD II connector has an attack path with a feasibility rating set to high since such an attack can be accomplished with low effort due to existing commercial 3-rd party OBD devices that are common. The feasibility rating in case of the cellular interface is also high, i.e., the attack path can be accomplished with low effort because the cellular interface are used for telematics and several attacks were already reported, e.g., [10], [22]. Hosted 3-rd party corrupted applications have a low feasibility rating because they require expertise and a corrupted provider while the automotive software market is well controlled. The USB port has a medium feasibility rating since USB sticks commonly carry unwanted software. The introduction of malicious software (malware) has a low feasibility since it requires expertise from multiple experts. Also, we score a low feasibility rating for the software or hardware development which is again subject to a mature development process. Vulnerabilities may still be possible due to the high software complexity, possibly reaching 8 million lines of code for a single ECU, and numerous companies, e.g., 8-11, working on the software for a single ECU [20]. A low feasibility is associated to the addition of a new unauthorized ECU and for software updates, i.e., corrupted software stacks on the ECU which evades detection to cause the attack, since this requires multiple experts to design. Additionally, for damage scenarios D8 and D10, another attack path can be considered, i.e., the manipulation of information collected by the sensors from the environment which can be at least ranked as having a medium feasibility. There is an increasing number of works that show clever manipulations of environmental data such as traffic signs [25], traffic lights [35], road lanes [30] and distances toward objects [33], [36].

We note that the attack path can be subject to a more complex feasibility analysis as done by the authors in [28]. They consider that the feasibility of an attack path is the product of probabilities associated to each edge from the path. The ISO 21434 however does not quantify feasibility as a probability and it was the choice of the authors from [28] to associate a probability to each rating, e.g., when the risk is high $p \in [0.9, 1]$ and when the risk is medium $p \in [0.5, 0.9)$.

Risk determination. According to ISO 21434 [1], risk values are determined based on the impact rating and the attack feasibility using the following relation:

$R = 1 + I \times F$, where R is the risk value, I is the impact rating and F is the feasibility rating. For the impact rating, in our calculation for the risk values we consider the maximum of the four types of impact (safety, financial, operational and privacy) which in this case is given by the safety component. This means that in case of the threats T1, T2, T3 which correspond to fuzzing, replays and stealthy attacks, the impact is major. While for a DoS the impact is only moderate. These values can be retrieved from Table 3. According to the standard, the four class impact, expressed as $\{negligible, moderate, major, severe\}$, is translated to numerical values as $\{0, 1, 1.5, 2\}$. The feasibility which is expressed as a four rank class $\{very\ low, low, moderate, high\}$ is translated to numerical values as follows $\{0, 1, 1.5, 2\}$. Consequently, a moderate impact incurs a numerical cost $I = 1$ and the severe impact corresponds to $I = 2$. A high feasibility ranking corresponds to $F = 2$ and consequently the impact ranking is $R = 1+2\times 2 = 5$. In Table 4 we depict the attack paths and the determined risk only for deceleration under the first two damage scenarios D1 and D2. For the rest of the signals and associated damages, risk determination should be done in a similar manner.

4.3 From determined risks to cybersecurity goals and concept

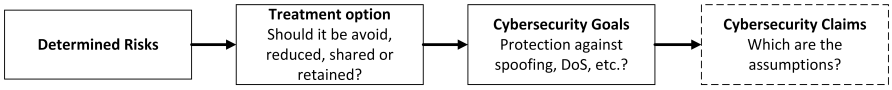


Fig. 14 Steps for product development according with to ISO 21434

In Figure 14 we illustrate the steps that follow from risk determination to integration and verification in case of cybersecurity attacks according to ISO 21434. Again, for an easier understanding of these steps, we formulate one question for each step which summarizes its expected outcome. According to ISO 21434 [1], the determined risks have to be treated in one of the following four ways: a) avoided by not starting or continuing a specific activity, b) reduced by using a proper security mechanism, c) shared, for example with insurances or d) retained. The cybersecurity goals are the result of the threat analysis and risk assessment (TARA) which we performed in the previous section. A cybersecurity goal, which results from the previous threat analysis, is a requirement to protect an assets against a threat according to ISO 21434 [1]. The cybersecurity claims must be formulated only in case of rationales for retaining or sharing the risks according to the same requirement. The claims can also include conditions for specific goals or functions for specific aspects, such as the use of a secure communication channel according to ISO 21434 [1].

Returning to the AEB system, in Table 5 we show the treatment option, cybersecurity goals and requirements for the threat scenarios on deceleration signal. For the

Table 5 Treatment option, cybersecurity goal and requirements for the threat scenarios on the deceleration signal

Threat scenario	Treatment Option	Cybersecurity Goal	Cybersecurity Requirement	
			Description	Allocation
T1. Fuzzing on deceleration	Reducing the risk	Deceleration shall be protected against spoofing by authentication or change detection mechanism	Verify if received data comes from a valid entity	All ECUs from attack path
			Implement change detection mechanism	AEB ECU
T2. DoS on deceleration	Reducing the risk	Deceleration shall be protected against DoS attacks by detection and recover the signal	Measures to detect and recover from a denial of service attack shall be employed	AEB ECU and Instrument cluster
T3. Replay on deceleration	Reducing the risk	Deceleration shall be protected against replay attacks by authentication (including the appropriate freshness parameters, timestamp)	Authentication (include strong time parameters, timestamp)	AEB ECU
T4. Stealthy on deceleration	Reducing the risk	Deceleration shall be protected against stealthy attacks by authentication	Verify if received data comes from a valid entity	All ECUs from attack path

rest of the assets (signals) in our analysis the details would be similar and we omit them for brevity.

Treatment. The treatment is the same in case of all assets: to reduce the risk. It is not acceptable to share the risk with an insurance company since they may result in fatal accidents. Clearly, the risk cannot be retained either, nor avoided by cancelling the AEB functionality which would contradict the main purpose of the system. Thus, the only treatment is to reduce the security risk. For this reason, the cybersecurity claims are not needed in our table. Cybersecurity claims are needed only when the treatment option is to retain or share the risk.

Cybersecurity goals and requirements. The decision to reduce the risk, moves us to the obvious goal to protect the signals, deceleration in particular, as outlined in Table 5, against spoofing, DoS, replay as well as against stealthy attacks.

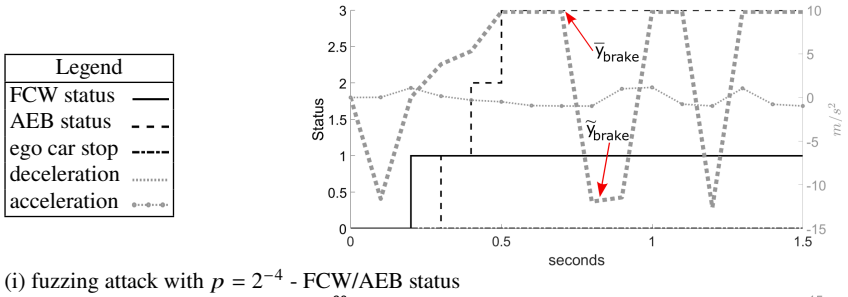
For fuzzing on deceleration we proposed two requirements: a) a change detection mechanism which needs to be implemented on the deceleration signal form the AEB ECU and b) a verification procedure enforced by cryptographic security for the received data, which needs to be implemented on all ECUs in order to check that CAN frames comes from a valid entity. The implementation of the second requirement should offer sufficient protection but it requires cryptographic capabilities that may be too expensive for some controllers and the use of a change detection mechanism may be cheaper and still provide some degree of protection. For DoS on deceleration, the cybersecurity requirement is to implement measures to detect and recover from a denial of service attack on the AEB ECU and instrument cluster. Authentication, implying the existence of strong time-variant parameters, i.e., timestamps, is needed against replay attacks. For the stealthy attacks, i.e, surge, bias and geometric, the requirement is to implement mechanisms to verify the source of the received data.

This description of the cybersecurity goals and requirements would be incomplete if we do not further give concrete suggestion on the exact mechanism that should be used for achieving these goals. Regarding intrusion detection, we have already pointed out a basic change detection mechanism. There are various other mechanisms that have been considered for intrusion detection in the literature including changes of specific parameters in heavy-duty J1939 vehicles [17], entropy analysis [23], [21] or Hamming distances possibly coupled with Bloom filters [13]. Other authors have proposed the use of precedence graphs [16], Markov models [24] or finite-state automatons [32].

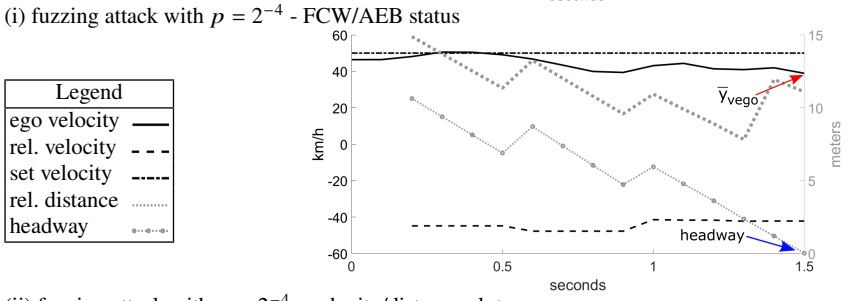
Regarding authentication mechanisms that validate the source of the frames, we have to leverage the discussion toward the AUTOSAR security standard for on-board communication [5]. According to the AUTOSAR SecOC [5] standard, the communication between two ECUs needs to be secured by authentication. In order to achieve this, the messages from the sender ECU contains a Protocol Data Unit (PDU) which helds the data and the timestamp or the freshness value (CNT) which is computed internally by the sender ECU and increases in time. Based on the PDU, including the freshness value (CNT), a cipher-based message authentication code (CMAC) is computed and the sender ECU transmits the PDU, CNT and CMAC to the receiver. Subsequently, the receiver ECU checks the CNT and if the CNT is correct it is used for the CMAC verification.

The AUTOSAR SecOC [5] standard specifies three security profiles on pages 62-63 that have to use 32-bit truncated CMAC-AES for authentication. Assuming that the secret key is secure, this would lead to a probability of 2^{-32} for an adversary to inject a valid frame (this is equivalent to a false negative event). However, the situation is much worse if we consider replays, not last correlated with stealthy manipulations, since the authentication tag of replayed frames is computed with the correct key. The only way to circumvent these attacks is with the proper freshness parameters which according to AUTOSAR SecOC [5] have 8 bits in profile 1, 0 bits in profile 2 and 4 bits in profile 3. This means that there are 256, 0 and 16 possible values for the time-variant parameter which is slightly low (or non-existent). At best, assuming an 8 bit counter, the probability of an injection would be $2^{-8} = 0.3\%$. The 4-bit length for the freshness parameter is too low for serious security demands.

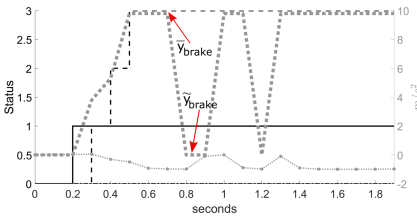
To illustrate the effectiveness of this layer of cryptographic protection, we chose the deceleration signal which had significant impact in case of fuzzing and stealthy attacks. We illustrate the behaviour in case of attacks with attack probability $p = 2^{-4}$ and $p = 2^{-8}$ that would result from using the corresponding freshness parameter on 4 or 8 bits. In Figure 15 we illustrate the signals under: (i), (ii) fuzzing attack with $p = 2^{-4}$, (iii), (iv) surge attack with $p = 2^{-4}$ and (v), (vi) fuzzing attack with $p = 2^{-8}$. When the attack probability is reduced to $p = 2^{-8}$, the AEB system is not affected by the adversarial signal $\widetilde{y}_{\mathsf{brake}}(k)$. In Table 6 we show the collision velocity and distance to target in case of fuzzing, and stealthy attacks on deceleration with attack probability $p = 2^{-4}$ and $p = 2^{-8}$. In case of an attack probability of $p = 2^{-4}$, the fuzzing attack causes an impact at significant velocity, the surge and bias attack cause impact at low velocity, while in case of the geometric attack no collision takes place. In case of an attack probability of $p = 2^{-8}$ no collision takes place. Also,



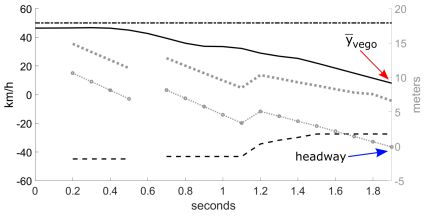
(i) fuzzing attack with $p = 2^{-4}$ - FCW/AEB status



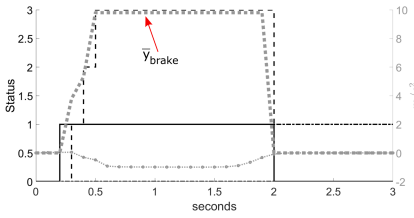
(ii) fuzzing attack with $p = 2^{-4}$ - velocity/distance plot



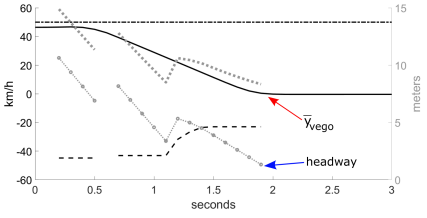
(iii) surge attack with $p = 2^{-4}$ - FCW/AEB status



(iv) surge attack with $p = 2^{-4}$ - velocity/distance plot



(v) fuzzing attack with $p = 2^{-8}$ - FCW/AEB status



(vi) fuzzing attack with $p = 2^{-8}$ - velocity/distance plot

Fig. 15 Signals under fuzzing and surge attacks on deceleration y_{brake} with $p = 2^{-4}$ and $p = 2^{-8}$: (i) fuzzing attack with $p = 2^{-4}$ - FCW/AEB status, (ii) fuzzing attack with $p = 2^{-4}$ - velocity/distance plot (iii) surge attack with $p = 2^{-4}$ - FCW/AEB status, (iv) surge attack with $p = 2^{-4}$ - velocity/distance plot, (v) fuzzing attack with $p = 2^{-8}$ - FCW/AEB status and (vi) fuzzing attack with $p = 2^{-8}$ - velocity/distance plot

by comparing with the results from Sections 3.3 and 3.4 the collision velocity is decreasing as the attack probability decreases, leading eventually to no impact when the attack probability is only $p = 2^{-8}$. The 8-bit freshness parameter is sufficient if we consider randomized injections with previously recorded frames, but it is too low for a more powerful adversary that records the order of the frames on the bus. For this reason, extending the 8-bit freshness parameter to a larger, 32 or 64-bit counter is needed, but this can only be achieved with the larger CAN-FD frames that have 512 bit datafields. We believe this is the only alternative for a high level of security.

Table 6 AEB results: collision velocity and distance to target in case of fuzzing, and stealthy attacks on deceleration with attack probability $p = 2^{-4}$ and $p = 2^{-8}$

	$p = 2^{-4}$		$p = 2^{-8}$	
Attack	**Collision velocity[km/h]**	**Distance to target[m]**	**Collision velocity[km/h]**	**Distance to target[m]**
Fuzzing	38.91	0	no coll.	1.32
Surge	7.99	0	no coll.	1.32
Bias	6.38	0	no coll.	1.32
Geometric	no coll.	0.92	no coll.	1.32

A further step is the cybersecurity validation at the vehicle level which is followed by the production of the actual item or component, i.e., clauses 11 and 12 of ISO 21434 [1]. These details are out of scope for the current presentation which was focused on the cybersecure-aware design alone. Lastly, specific operation and maintenance activities, suggested in clause 13 of ISO 21434 [1], will also occur during the vehicle lifetime which may also lead to re-designs of the cybersecurity goals and claims.

5 Conclusion

Two lines of defence are advocated by our analysis. One of them is the inclusion of intrusion detection systems, such as the basic change detection outlined in our analysis. This line of defense requires a careful selection of specific parameters, i.e., thresholds and biases, which have to be the subject of careful engineering maturity testing and verification which are not fully possible in our work. The second line of defense is the adoption of cryptographic security that will ensure that each frame is authentic and, similarly important, fresh in order to remove the possibility of a replay attack. For this, using regular 64 bit CAN frames has its limit. More specifically, in accordance to AUTOSAR SecOC [5], at most 8 bits are used as freshness parameter which offers only a limited protection against replay attacks. For this reason, we believe that the adoption of CAN-FD which extends the datafield to 512 bits and allows a larger freshness parameter, such as a 64 bit timestamps as commonly available in network synchronization protocols, is the only way to ensure that freshness and thus complete source authentication is achieved.

References

1. ISO/SAE DIS 21434: Road vehicles — cybersecurity engineering. 2020.
2. Addendum 154 – UN regulation no. 155: Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system. 2021.
3. Test protocol – aeb vru systems, version 3.0.3. In *Vulnerable Road User (VRU) Protection*. Euro NCAP, June 2020.
4. AUTOSAR. *Specification of Intrusion Detection System Protocol*, r20-11 edition, 2020.
5. AUTOSAR. *Specification of Secure Onboard Communication*, r20-11 edition, November 2020. no. 654.
6. L. ben Othmane, R. Ranchal, R. Fernando, B. Bhargava, and E. Bodden. Incorporating attacker capabilities in risk estimation and mitigation. *Computers & Security*, 51:41–61, 2015.
7. T. Brandt and T. Tamisier. The future connected car–safely developed thanks to unece wp. 29? In *21. Internationales Stuttgarter Symposium*, pages 461–473. Springer, 2021.
8. M. Brown. Addressing the challenges of a sector in transformation and preparing to meet new cyber compliance requirements (ISO/SAE 21434). BSI Group, 2022.
9. A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 355–366, 2011.
10. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium (USENIX Security 11)*, 2011.
11. S. Gautham, A. V. Jayakumar, and C. Elks. Multilevel runtime security and safety monitoring for cyber physical systems using model-based engineering. In *International Conference on Computer Safety, Reliability, and Security*, pages 193–204. Springer, 2020.
12. B. Groza, H.-E. Gurban, and P.-S. Murvay. Designing security for in-vehicle networks: a body control module (bcm) centered viewpoint. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 176–183. IEEE, 2016.
13. B. Groza and P.-S. Murvay. Efficient intrusion detection with bloom filtering in controller area networks. *IEEE Transactions on Information Forensics and Security*, 14(4):1037–1051, 2018.
14. E. H. Gurban, B. Groza, and P.-S. Murvay. Risk assessment and security countermeasures for vehicular instrument clusters. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 223–230. IEEE, 2018.
15. O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl. Security requirements for automotive on-board networks. In *2009 9th International Conference on Intelligent Transport Systems Telecommunications,(ITST)*, pages 641–646. IEEE, 2009.
16. R. Islam, R. U. D. Refat, S. M. Yerram, and H. Malik. Graph-based intrusion detection system for controller area networks. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
17. C. Jichici, B. Groza, R. Ragobete, P.-S. Murvay, and T. Andreica. Effective intrusion detection and prevention for the commercial vehicle sae j1939 can bus. *IEEE Transactions on Intelligent Transportation Systems*, 2022.
18. P. Kapoor, A. Vora, and K.-D. Kang. Detecting and mitigating spoofing attack against an automotive radar. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–6. IEEE, 2018.
19. G. Macher, C. Schmittner, O. Veledar, and E. Brenner. ISO/SAE DIS 21434 automotive cybersecurity standard-in a nutshell. In *International Conference on Computer Safety, Reliability, and Security*, pages 123–135. Springer, 2020.
20. R. Mader, G. Winkler, and N. Reindl, Thomas amd Pandya. The car's electronic architecture in motion: The coming transformation. In *42nd International Vienna Motor Symposium*, 2021.
21. M. Marchetti, D. Stabili, A. Guido, and M. Colajanni. Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. In *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pages 1–6. IEEE, 2016.

22. C. Miller and C. Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015(S 91), 2015.
23. M. Müter and N. Asaj. Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 1110–1115. IEEE, 2011.
24. S. N. Narayanan, S. Mittal, and A. Joshi. Obd_securealert: An anomaly detection system for vehicles. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–6. IEEE, 2016.
25. B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici. *Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks*, page 293–308. Association for Computing Machinery, New York, NY, USA, 2020.
26. S. Nie, L. Liu, and Y. Du. Free-fall: Hacking tesla from wireless to can bus. *Briefing, Black Hat USA*, 25:1–16, 2017.
27. C. Plappert, D. Zelle, H. Gadacz, R. Rieke, D. Scheuermann, and C. Krauß. Attack surface assessment for cybersecurity engineering in the automotive domain. In *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 266–275. IEEE, 2021.
28. D. Püllen, J. Liske, and S. Katzenbeisser. ISO/SAE 21434-based risk assessment of security incidents in automated road vehicles. In *International Conference on Computer Safety, Reliability, and Security*, pages 82–97. Springer, 2021.
29. K. Razikin and B. Soewito. Cybersecurity decision support model to designing information technology security system based on risk analysis and cybersecurity framework. *Egyptian Informatics Journal*, 2022.
30. T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen. Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3309–3326, 2021.
31. C. Schmittner, B. Schrammel, and S. König. Asset driven ISO/SAE 21434 compliant automotive cybersecurity analysis with threatget. In *European Conference on Software Process Improvement*, pages 548–563. Springer, 2021.
32. I. Studnia, E. Alata, V. Nicomette, M. Kaâniche, and Y. Laarouchi. A language-based intrusion detection approach for automotive embedded networks. *International Journal of Embedded Systems*, 10(1):1–12, 2018.
33. J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 877–894, 2020.
34. Y. Wang, Y. Wang, H. Qin, H. Ji, Y. Zhang, and J. Wang. A systematic risk assessment framework of automotive cybersecurity. *Automotive Innovation*, 4(3):253–261, 2021.
35. C. Yan, Z. Xu, Z. Yin, X. Ji, and W. Xu. Rolling colors: Adversarial laser exploits against traffic light recognition. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association.
36. C. Zhou, Q. Yan, Y. Shi, and L. Sun. DoubleStar: Long-Range attack towards depth estimation based obstacle avoidance in autonomous systems. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, Aug. 2022. USENIX Association.