Intelligent Transportation

Integrated safety and security enhancement of connected automated vehicles using DHR architecture

Yufeng Li\textsuperscript{1,2,⋆}, Qi Liu\textsuperscript{1}, Xuehong Chen\textsuperscript{3}, and Chenhong Cao\textsuperscript{1,2,⋆}

\textsuperscript{1} School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
\textsuperscript{2} Purple Mountain Laboratories, Nanjing 211111, China
\textsuperscript{3} China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, China

Received: 21 January 2022 / Accepted: 25 November 2022 / Published online: 27 January 23

Abstract Safety and security are interrelated and both essential for connected automated vehicles (CAVs). They are usually investigated independently, followed by standards ISO 26262 and ISO/SAE 21434, respectively. However, more functional safety and security features of in-vehicle components make existing safety mechanisms weaken security mechanisms and vice versa. This results in a dilemma that the safety-critical and security-critical in-vehicle components cannot be protected. In this paper, we propose a dynamic heterogeneous redundancy (DHR) architecture to enhance the safety and security of CAVs simultaneously. We first investigate the current status of integrated safety and security analysis and explore the relationship between safety and security. Then, we propose a new taxonomy of in-vehicle components based on safety and security features. Finally, a dynamic heterogeneous redundancy (DHR) architecture is proposed to guarantee integrated functional safety and cyber security of connected vehicles for the first time. A case study on an automated bus shows that DHR architecture can not only detect unknown failures and ensure functional safety but also detect unknown attacks to protect cyber security. Furthermore, we provide an in-depth analysis of quantification for CAVs performance using DHR architecture and identify challenges and future research directions. Overall, integrated safety and security enhancement is an emerging research direction.

Keywords Connected automated vehicles, Safety, Security, Dynamic heterogeneous redundancy (DHR), Quantification

Citation Li Y, Liu Q and Chen X et al. Integrated safety and security enhancement of connected automated vehicles using DHR architecture. Security and Safety 2023; 2: 2022009. https://doi.org/10.1051/sands/2022009

1 Introduction

The integration of electrical and electronic systems with communication technologies, such as over-the-air (OTA), Telematics, and vehicle-to-vehicle (V2X), gave rise to automotive industry. As a result, the emerging connected vehicles embracing more powerful perception and behavioral capabilities facilitate many interesting and convenient services for driving. However, the increased level of connectivity and automation also brings undesirable events: failures (safety) and intentional attacks (security).

It is well recognized that vehicles are safety-critical systems whose failure can lead to injuries and loss of life. For a long time, tremendous efforts are focused on vehicle safety concerns that have large impacts on the environment and even threaten human lives. However, only part of the accidental component failures

* Corresponding authors (email: liyufeng.shu@shu.edu.cn (Yufeng Li); caoch@shu.edu.cn (Chenhong Cao))

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

© The Author(s) 2023. Published by EDP Sciences and China Science Publishing & Media Ltd.
or software errors were traditionally addressed. Today, the functional safety of automated vehicles is facing more severe challenges due to the rapid increase in the amount of code. A modern autonomous car now contains more than 100 million lines of software code and is expected to have around 300 million lines of code by 2030. In fact, mature software development teams produce two to five bugs per thousand lines of code.

As automated vehicles become increasingly connected, security risks are increasing because of the potential for deliberate harm by adversaries. The attack surface includes short-range and long-range automotive wireless interfaces such as bluetooth, remote keyless entry, RFIDs, WIFI, global positioning system (GPS), satellite radio, etc. [1]. As an example, vulnerabilities in the application interface of an on-board diagnostics (OBD) dongle can allow an attacker to inject malicious code into it [1]. Another example is that a compromised compact disc (CD) player can offer an effective vector for attacking other automotive components since many automotive media systems are now controller area network (CAN) bus interconnected [1].

As a result, researchers have become increasingly aware of the security-related risks that threaten automated vehicles. Different from the traditional cyber security on the Internet, the security of autonomous vehicles is more critical since it may threaten human lives. Considerable research effort is being invested in identifying cybersecurity vulnerabilities, recommending potential mitigation techniques, as well as highlighting the knowledge discrepancies that can be used as a guideline to address the cybersecurity problems in connected vehicles [2–4].

In fact, safety and security are interrelated and both essential for automated vehicles. Typical electrical/electronic (E/E) architecture categorizes components by functions, such as perception, decision-making, communication, control, and execution. Functional safety and security features of in-vehicle components are ignored. As a result, safety-critical and security-critical in-vehicle components cannot be protected by existing safety mechanisms and security mechanisms. For example, the advanced driving assistance system (ADAS) is a typical safety mechanism that can reduce accidents and injuries significantly. However, it has also become a high-value target for cyberattacks, which aggravates the security problem. Another example is the security mechanism such as the firewall or intrusion detection system (IDS) which is typically employed as part of in-car communication. The design flaws of these mechanisms may cause malfunctions, interrupting the in-car network and causing safety accidents.

Nowadays, many researchers of co-engineering safety and security are trying to mitigate the risk caused by accident malfunctions or malicious attacks. Since connected automated vehicles are safety-critical systems, it is necessary to satisfy safety and security simultaneously as they can affect each other. However, safety and security techniques are developing independently. Consequently, researchers are increasingly interested in how techniques from safety would complement or conflict with those from security.

To enhance the safety and security of connected automated vehicles (CAVs) simultaneously, we propose a novel generalized robust control technology – DHR architecture, which can not only detect unknown failures and ensure functional safety, but also detect unknown attacks to protect cyber security. The contributions of this paper can be summarized as:

1. We investigate the current status of integrated safety and security analysis and explore the relationship between safety and security.
2. We propose a new taxonomy of in-vehicle components based on safety and security features, which is helpful in developing joint safety and security enhancement technology.
3. We implement a prototype of DHR on an automated bus and conducted two test cases to validate the effectiveness of the DHR architecture when facing functional failures and cyberattacks.
4. We provide an in-depth analysis of quantification for CAVs performance using DHR architecture and point out some challenges and future research directions. This is non-trivial as quantitative safety and security analysis is pivotal for effective safety and security management.

---

2 Background

To solve conflicts between safety mechanisms and security mechanisms, we first identify inter-relationships between safety and security. Then, we review existing safety mechanisms and security mechanisms, and the corresponding weaknesses are summarized.

2.1 Inter-relationships between safety and security in CAVs

For CAVs, safety aims at protecting the vehicle from accidental failures to avoid hazards, and security focuses on protecting the vehicle from intentional attacks [5]. Both safety and security are related to the risk of electrical and electronic systems in CAVs. To better illustrate the distinction between safety and security, we describe them as a conceptual grid representing the two aspects in Figure 1.

- Security risks are malicious. Attackers may gain unauthorized access to the vehicle’s function through mobile apps remotely and control vehicles to damage the environment or threaten human life. Moreover, attackers can deceive the CAV to make wrong judgments by projecting or setting fake environmental information such as a fake pedestrian or lane markers projected on a road by a projector-equipped drone [6].
- Safety risks are accidental. On the one hand, accidental components failure of CAVs may cause serious damage to the environment. On the other hand, extreme weather such as heavy rain or snow may cause malfunctions in automated vehicles.

In the past, vehicles could only rely on reliability technologies such as active safety mechanisms and passive safety mechanisms to guarantee safety. However, the safety of today’s CAV not only relies on traditional safety technology but also depends on cyber security technology.

2.2 Safety mechanisms in CAVs

Traditionally, vehicles rely on reliable technologies such as active safety mechanisms and passive safety mechanisms to guarantee safety. Active schemes aim to prevent vehicles from crashing, such as driving assistance schemes including automated braking [7], backup camera [8], adaptive headlamps [9], lane departure warning systems [10], etc. Passive schemes aim to protect the driver and passengers from crash injuries, such as the airbag, crumple zone, headrest, seat belt, and a laminated windshield.
However, the safety of CAV not only relies on traditional safety technology but also depends on cybersecurity technology. For example, in 2015, two American hackers attacked a Jeep. The important functions such as engine and brakes are remotely taken over via mobile phone network [11]. They controlled the accelerator to let the Jeep stop on the highway leaving the driver in a rather dangerous situation. Therefore, the cybersecurity of CAV is also critical for safety.

Conventional vehicles mostly focus on functional safety for mechanical failure. With the increasing automation and connectivity of CAVs, more efforts are needed to identify the safety risks raised by the software failures of the communication components and the autonomous driving components and propose appropriate defense mechanisms.

### 2.3 Security mechanisms in CAVs

Security attacks on CAVs include attacks on networks and attacks on the vehicle itself. Many research works have been carried out, focusing on a particular kind of security attack such as a CAN attack [12]. Traditional security technologies such as authentication [13], detection [14], and cryptography [15] are usually employed to deal with these attacks. To our knowledge, these technologies usually need prior knowledge and cannot deal with unknown vulnerabilities such as 0-day attacks [16] in real time.

As described previously, the security and safety of CAVs are interrelated, e.g., security attacks can result in CAV functional failures and cause safety problems. It is not hard to imagine the potential destruction of the environment and property when the CAV falls into the wrong hands through cyberattacks. Thus, mechanisms that jointly consider safety and security are desirable.

### 2.4 Integrated safety and security mechanisms in CAVs

Due to the importance and relevance of safety and security in CAVs, several studies have recently emerged that aim to identify, assess, and manage risks related to both safety and security. These studies can be classified based on the overall goal [17]:

- **Security-informed safety approaches**: Methods that incorporate security techniques into safety techniques to achieve a safe system.
- **Safety-informed security approaches**: Methods that incorporate safety techniques into security techniques to achieve a secure system.
- **Combined safety and security approaches**: Methods that combine safety techniques and security techniques to achieve a both safe and secure system.

Standard SAE J3061 [18] is a cyber security guidebook for vehicle systems that also provides a way to incorporate the process of functional safety standard ISO 26262. SAHARA [19] and US2 [20] further investigate safety and security issues and choose appropriate countermeasures. However, such process integration cannot change the “add-on” characteristic of the security defense technology. The “add-on” security technologies will not only increase the software code of CAVs and bugs to increase the potential safety risks but also bring computational overhead.

Recently, many researchers proposed approaches of safety and security co-engineering to harmonize the conflicts [17] between safety mechanisms [7–10] and security mechanisms [13–15]. Table 1 describes the characterization of recent studies in safety and security co-engineering for CAVs. More details can be referred to [21]. The model depicts the approach on which the analysis is based, including graphical, formal, and both graphical and formal [22]. The lifecycle explains the approach adopted in which phase of the system lifecycle, such as requirement (RE), risk analysis (RA), and any phase-generic (GE) [23]. Conflict resolution means the approach facilitates the identification and study of potential conflicts between safety and security aspects.

Survey on security and safety co-engineering for CAVs shows the conclusion as follows [21]:

- Few methods are in compliance with safety and security standards. Automated vehicles operating on the road must follow safety and security standards.
- Lack of quantitative approaches. It is well known that analyzing security threats quantitatively is a challenge in most cases in the real world. Therefore, combining quantitative and qualitative methods for safety and security co-engineering is worth exploring.
### Table 1. Characterization

<table>
<thead>
<tr>
<th>Study</th>
<th>Model</th>
<th>Associate with a standard</th>
<th>Goal</th>
<th>Lifecycle</th>
<th>Conflict</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>F</td>
<td>No</td>
<td>Both</td>
<td>RE</td>
<td>N</td>
</tr>
<tr>
<td>[24]</td>
<td>G</td>
<td>Yes</td>
<td>Both</td>
<td>RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[25]</td>
<td>G</td>
<td>Yes</td>
<td>Safety</td>
<td>RA</td>
<td>N</td>
</tr>
<tr>
<td>[26]</td>
<td>G</td>
<td>Yes</td>
<td>Both</td>
<td>RA</td>
<td>N</td>
</tr>
<tr>
<td>[27]</td>
<td>Both</td>
<td>No</td>
<td>Safety</td>
<td>RE</td>
<td>N</td>
</tr>
<tr>
<td>[28]</td>
<td>F</td>
<td>Yes</td>
<td>Safety</td>
<td>RE</td>
<td>N</td>
</tr>
<tr>
<td>[29]</td>
<td>F</td>
<td>Yes</td>
<td>Safety</td>
<td>RA</td>
<td>N</td>
</tr>
<tr>
<td>[31]</td>
<td>Both</td>
<td>Yes</td>
<td>Both</td>
<td>GE/RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[32]</td>
<td>G</td>
<td>Yes</td>
<td>Both</td>
<td>RE</td>
<td>Y</td>
</tr>
<tr>
<td>[33]</td>
<td>G</td>
<td>Yes</td>
<td>Safety</td>
<td>RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[34]</td>
<td>G</td>
<td>Yes</td>
<td>Safety</td>
<td>GE/RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[35]</td>
<td>F</td>
<td>No</td>
<td>Both</td>
<td>GE/RA</td>
<td>N</td>
</tr>
<tr>
<td>[36]</td>
<td>F</td>
<td>No</td>
<td>Both</td>
<td>GE/GE/GA</td>
<td>N</td>
</tr>
<tr>
<td>[37]</td>
<td>F</td>
<td>No</td>
<td>Security</td>
<td>RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[38]</td>
<td>F</td>
<td>No</td>
<td>Security</td>
<td>RE/RA</td>
<td>N</td>
</tr>
<tr>
<td>[40]</td>
<td>G</td>
<td>No</td>
<td>Safety</td>
<td>RA/RE</td>
<td>N</td>
</tr>
<tr>
<td>[41]</td>
<td>G</td>
<td>No</td>
<td>Safety</td>
<td>RA</td>
<td>N</td>
</tr>
</tbody>
</table>

- Lack of guidance on resolving conflicts between safety and security mechanisms. This is a challenge worth studying.

Different from them, we propose a novel mechanism that can effectively guarantee functional safety and cyber security simultaneously. The proposed method can not only deal with known vulnerabilities, but also unknown vulnerabilities.

### 3 A New taxonomy for in-vehicle components

To develop a joint safety and security mechanism, the typical electrical/electronic (E/E) architecture for CAV is first analyzed. Then, a new taxonomy of in-vehicle components is proposed based on their safety/security attributes, which means the in-vehicle components would suffer from what kind of risk, such as accidental failures, or intentional attacks. This is especially helpful in developing the joint mechanism.

#### 3.1 Typical E/E architecture for CAV

Figure 2 shows a typical CAV system architecture, which is composed of four kinds of components.

- Perception components: This layer includes different perception components used to obtain environmental information, such as vision sensor, LIDAR, Millimeter-Wave RADAR, ultrasonic sensor, and infrared sensor.
- Decision-making components: These components perform as the brain of an automated vehicle. It is important to make critical decisions based on the driving environment. Functions like sensor fusion, path planning, semantic understanding, positioning, and tracking are provided for better decision-making.
- Communication components: This layer includes different components used for inter-vehicle and intra-vehicle communication. For example, Telematics BOX (T-BOX), CAN network, and in-vehicle gateway are typical communication components.
- Control and execution components: These components are responsible for controlling the electronic subsystems, such as the drive-by-wire system that consists of steering, braking, accelerator, gears, and intelligent cockpit.
3.2 Safety/security attributes of in-vehicle components

Since traditional taxonomy categorizes components by functions, functional safety and security features of in-vehicle components are ignored. As a result, safety-critical and security-critical in-vehicle components cannot be protected by existing safety mechanisms and security mechanisms. Thus, a new taxonomy of in-vehicle components based on safety/security attributes is proposed, which contributes to the development of a joint safety and security mechanism. As illustrated in Figure 3, we define three types of safety/security attributes and classify the in-vehicle components as follows.

- **Pure safety-critical components** are safety-related components including mechanical components and E/E components that are unreachable by cyberattacks. Mechanic components such as the steering wheel, brake pedal, accelerator pedal, and gearshift lever are pure safety-critical components. Some electronic control units (ECU) can also be regarded as pure safety-critical components when cyberattacks are unreachable.

- **Pure security-critical components** are security-related components that are reachable by cyberattacks and are isolated from safety domains such as the power domain and control domain. The highly isolated entertainment system is usually a pure security-critical component. Cyberattacks on this component usually result in damage to the confidentiality of information but will not affect the safety of CAVs.

- **Safety-critical & security-critical components** are safety-related components that are reachable by cyberattacks. We argue that the ADAS and the in-vehicle communication components, such as Telematics BOX (T-BOX), are the safety-critical & security-critical components. Failure of these components, as well as cyberattacks on these components, can cause accidents and casualties. With the development of CAVs, the scope of safety-critical & security-critical components would constantly expand.

Pure safety-critical components and pure security-critical components can be protected by existing safety and security mechanisms. However, for the safety-critical & security-critical components, novel mechanisms should be developed to effectively guarantee safety and security at the same time. In the following section, a dynamic heterogeneous redundancy (DHR) architecture is proposed for CAVs to achieve both safety and security.
No fatality in the aviation industry has been for over a quarter of a century. The architectures in aviation relied on hardware and process redundancy. For example, safety-critical aircraft FBW systems use masking, redundancy, and reconfiguration to maintain normal operation after a failure [42]. As for automated vehicle systems, they can be cheaper and less sophisticated by designing a Fail-Operational architecture. Inspired by this, we propose a DHR architecture for CAVs.

4 DHR architecture-new exploration of generalized robust control technology

4.1 DHR architecture for CAV

The basic concept of DHR is the “relative correctness” axiom. That is, any system has a variety of software and hardware flaws. When multiple completely heterogeneous systems perform the same task at the same time, in the same place, the possibility of failure caused by the same flaw is extremely low. Therefore, the most consistent results of the multiple heterogeneous systems are relatively correct.

Based on the “relative correctness”, multiple heterogeneous executors can be employed to accomplish the same CAV component function. According to [43], for two executors with sufficient heterogeneity, the probability that they failed for the same flaw is generally $1 \times 10^{-4}$ or less. For example, with three heterogeneous executors, when executor A failed at a certain moment, the probability that the other two executors also failed with the same flaw is extremely low. When inconsistency of executors’ output is detected, the consistent output of most executors is taken as the correct output and the abnormal executor is detected. The above judgment process is called a “consensus mechanism”. Moreover, the abnormal executors will be replaced by the normal executors with the same function.

The defense system based on the DHR architecture is shown in Figure 4, which contains an input agent, executor set, arbiter, feedback controller, and component pool.

The input agent distributes tasks to the executor set which consists of multiple heterogeneous executors with the same function. For CAVs, the perception and decision unit (autopilot module) can be chosen as an executor, which performs key and fundamental functions in safety-critical & security-critical components.

The arbiter judges the content consistency of executors’ outputs according to the consensus mechanism. The feedback controller determines whether to send an instruction to the input agent and chooses
a normal executor from the component pool to replace the abnormal executor. The component pool consists of multiple functionally equivalent heterogeneous executors.

With the DHR architecture, integrated safety and security can be obtained. In CAVs, safety risks are usually raised by design defects and security threats are usually raised by vulnerability. Both the design defects and the vulnerabilities can cause abnormal behaviors in executors. With the DHR architecture, abnormal executors can be detected by the consensus mechanism and the system can restore to normal states by reconstructing abnormal executors dynamically.

Moreover, using the DHR architecture based on a consensus mechanism, unknown design defects and vulnerabilities can also be detected as long as they cause abnormality.

4.2 A practical DHR prototype for CAVs

A prototype has been developed based on the DHR architecture, where three L2 [44] advanced driver-assistance systems (ADASs) are employed as executors and one L2 ADAS is employed as a component pool. As shown in Figure 5, the first L2 executor consists of lidars, cameras, hardware, and software computing, and is implemented by Infineon platforms. The second L2 executor is implemented by an FPGA platform, including cameras and radars. The third L2 executor is implemented by a Freescale platform, including cameras and radars. The DHR prototype has been deployed on the “All Star” autonomous electric minibus.

The above heterogeneous executors are employed for the following reasons:

1. The L2 ADAS is the most common automated driving system in the market, with many heterogeneous suites available.
2. The price of L2 ADAS is an order of magnitude lower than that of L3 and L4 automated driving systems.

As to the arbiter, it was developed on a customized industrial control computer, running on Ubuntu 18.04 OS, equipped with Intel Core i5-6500 2.5 GHz x 4 CPU. The arbiter’s functions are programmed with C language. In this arbiter, three L2 ADAS were connected via the CAN bus.

In the practical development process, various environmental factors need to be considered to design a consensus mechanism. The consensus mechanism proposed in Algorithm 1.

In order to avoid false alarms caused by inconsistency in the perception range of each executor, the public perception field A is initialized based on experience and vehicle speed. Next, the consensus
mechanism is divided into perception arbitration and decision arbitration. Perception arbitration aims to judge whether each executor perceives an obstacle. Decision arbitration decides what kind of instructions the car should follow, such as acceleration, braking, turning left, or turning right.

In the perception arbitration, based on the obstacle detection results of executors, a calculation method of perceptual similarity is empirically proposed. The perceptual similarity between the executors is calculated in pairs. If the similarity is lower than the threshold, it is considered that the perception is inconsistent. And if the similarity is greater than the threshold, the perception is considered consistent. Finally, if the perception results among the three executors are consistent, the feedback is normal, otherwise, the inconsistency will be added to the cache queue.

In the next decision-making decision, the decision-making similarity between the executors is calculated pairwise. If the similarity is lower than the threshold, the decision is considered inconsistent. And if the similarity is greater than the threshold, the decision is considered consistent. Finally, if the perception results among the three executors are consistent, the feedback is normal, and the inconsistency will be added to the cache queue. Based on the above, to distinguish whether the current inconsistency is a normal fluctuation or an attack/fault, the cumulative inconsistency times in the recent historical time (1s) are analyzed based on the cache queue. If the inconsistency times are greater than the threshold, it is considered that an abnormality is currently found. Otherwise, it is assumed that no exception has occurred.

4.3 Evaluation

Two tests have been conducted to validate the DHR. The tests were set for 5 min. The arbiter detected the abnormality and made a decision every 100 ms, and 3000 decisions were made in 5 min. The comparison was one executor in working.

Test 1: This test aims to validate the effectiveness of the DHR architecture when facing functional failures. One obstacle was set on the road. In this test, E3’s instruction message is blocked to simulate the executor’s functional failure.
Algorithm 1: overall algorithm of a consensus mechanism

Input: executors outputs \( X = \{(p_1, d_1), (p_2, d_2), (p_3, d_3)\} \)
Vehicle speed \( v \)
Exception meta queue \( \text{cache}_\text{queue} \)

Output: arbitration result (normal or abnormal)

Initialization: similarity threshold \( T \), public perception area \( A \)

01: preprocess perception results \( \{p_1, p_2, p_3\} \) according to \( A \)
02: for all \( i \in \{1, 2, 3\} \) do # calculate the output perception similarity in pairs
03: compute \( \text{perception\_similarity}(p_i, p_j), j=(i+1)\text{mod}3 \)
04: if \( \text{perception\_similarity}(p_i, p_j) < T \), then
05: perception of executors \( i \) and \( j \) are inconsistent
06: else
07: perception of executors \( i \) and \( j \) are consistent
08: end if
09: end for
10: if the three executors’ perceptions are not consistent with each other
11: insert one perception exception to \( \text{cache}_\text{queue} \)
12: end if
13: calculate fused perception \( P \) according to \( \{p_1, p_2, p_3\} \)
14: judge whether braking is required according to \( P \) and \( v \)
15: if braking is required then
16: for all \( i \in \{1, 2, 3\} \) do # calculate the output decision similarity in pairs
17: compute \( \text{decision\_similarity}(d_i, d_j), j=(i+1)\text{mod}3 \)
18: if \( \text{decision\_similarity}(d_i, d_j) \neq 100\% \), then
19: the decision of executors \( i \) and \( j \) are inconsistent
20: else
21: the decision of executors \( i \) and \( j \) are consistent
22: end if
23: end for
24: if three executors’ decisions are not consistent with each other
25: insert one decision exception to \( \text{cache}_\text{queue} \)
26: end if
27: obtain the final arbitration result according to the \( \text{cache}_\text{queue} \) analysis

Figure 6. The arbitration process of DHR architecture under functional failure (one obstacle)

Figure 6 shows the arbitration process of DHR architecture under functional failure. The red border represents the number and location information of obstacles perceived by executors. E1 and E2’s perception results reported that there was one obstacle ahead, and suggested braking. However, E3’s perception was empty due to the functional failure. The arbiter compared the output of executors every 100 ms and chose the consistent output of the most executors as the final output. As shown in the box with a yellow border in Figure 6, the arbiter detects the abnormality of executors, and E3 would be replaced with a normal one from the component pool. Therefore, with the DHR architecture, the obstacle can be detected and the bus will get around the obstacle, as shown in Figure 7a. However, for the comparing system with only E3, zero obstacles will be detected due to the functional failure, and the comparing system will make a wrong decision, and the bus hit the obstacle, as shown in Figure 7b.

Test 2: This case validates the effectiveness of the DHR architecture when facing cyberattacks. The adversarial sensor attack on LiDAR-based perception was carried out on executor 3. The attack goal was set as spoofing obstacles close to the front of the “All Star” bus. OSRAM SFH 213FA, OSRAM SPL PL90, PCO-7114, and AFG3251 were used to inject spoofed points into the LiDAR sensor [45].
Figure 7. The field test under functional failure (one obstacle): (a) One executor; (b) DHR architecture

Figure 8. Adversarial sensor attack on LiDAR-based perception

Figure 9. The arbitration process of DHR under cyberattack (one fake obstacle)

Figure 8 shows the process of adversarial sensor attacks on LiDAR-based perception. The adversaries spoof a fake obstacle in front of a victim’s automated car by strategically transmitting laser signals to the victim’s LiDAR sensor.

Figure 9 shows the arbitration process of the DHR architecture under cyberattack. The contents in the red border include the number and location information of obstacles perceived by executors. It is shown that the perceptions of E1 and E2 were not spoofed since they are not based on LiDAR and their decisions were consistent. However, the perception of E3 was inconsistent with the other two, which reported that there was one obstacle ahead, and suggested braking. As shown in the box with a yellow border, with the DHR architecture, the arbiter can detect the abnormality of E3. So E3 would be replaced with a normal one from the component pool, and the attack no longer took effect. As a result, with the DHR architecture, the perception results collected from the executor set reported that there was no obstacle ahead, and the decision was to keep moving forward, as shown in Figure 10a. However, for the comparing system with only E3, the abnormality of E3 cannot be detected, so the comparing system will make a wrong decision, and the bus braked and stopped, as shown in Figure 10b.

Table 2 shows the experiment results for the detection success rate when the minimum confirming time is 300 ms. The correct decision success rate means DHR architecture makes the correct decision
Table 2. Study results on the detection success rate

<table>
<thead>
<tr>
<th>Failure config.</th>
<th>Correct decision success rate</th>
<th>Abnormal executor detection success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional fail</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Cyberattacks</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>100 ms</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>200 ms</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>300 ms</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>400 ms</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>500 ms</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

...to make the system in normal operation. Abnormal executor detection success rate means whether the executor in failure can be detected. As shown, both the two success rates can achieve 100% for functional and cyberattacks. Since the duration of the two failures can be considered to be infinite, the abnormality can be detected through the consensus mechanism. The abnormal executor will be replaced after three consecutive confirmations. Natural factors refer to the robustness of the obstacle detection algorithm of heterogeneous executors or the temporary shielding on the vehicle camera caused by the wiper (raindrops, mosquitoes). The duration of the executor’s failure is finite under natural factors. Note that the Abnormal executor detection success rate is zero when the failure duration of natural factors is less than 300 ms. Nevertheless, the DHR architecture can always make the correct decision by choosing the consistent output of the most executors as the final output.

The above cases validate the effectiveness of the DHR architecture. These cases consider the circumstance that only one executor failed due to the functional failure or the cyberattack. Note that the probability that multiple executors failed at the same time is extremely low due to the heterogeneity and dynamics of the DHR. In fact, the probability that multiple executors fail due to the same defect approaches 0 when the degree of heterogeneity is large enough.

Through this DHR architecture, the “All Star” bus can not only detect unknown failures and ensure functional safety, but also detect unknown attacks to protect cyber security.

5 Potential research directions for quantifiable validation of generalized robust control technology

In functional safety, hardware failures generally refer to random failures, which can be quantified by traditional reliability theory. Software failures in functional safety are systematic failures that cannot be directly quantified. Risk in functional safety can often be viewed as a function of failure, consequence, and probability of occurrence. Risk in cybersecurity may be viewed as a function of the severity of cyber threats, the vulnerability of the target system, the consequences of an attack, and the probability of an attack occurring. Risk quantification in cybersecurity is much more difficult than in functional safety.
Table 3. Estimated fatality rates for automated vehicles

<table>
<thead>
<tr>
<th>Case</th>
<th>Reliability compared with a benchmark</th>
<th>Failure rate (per hour)</th>
<th>Annual fatality</th>
<th>Fatality/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Same as</td>
<td>$5 \times 10^{-7}$</td>
<td>3650</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>$10\times$</td>
<td>$5 \times 10^{-8}$</td>
<td>365</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>$100\times$</td>
<td>$5 \times 10^{-9}$</td>
<td>365</td>
<td>1</td>
</tr>
</tbody>
</table>

Safety and security domains both try to make risk assessments quantitatively through probability theory. Cyber security threats are often expressed as estimated property damage, while safety hazards are the possibility that leads to accidents [4]. However, rarely co-analyzing the safety and security of automated vehicles quantitatively.

As discussed in safety and security co-engineering, it’s challenging to quantify safety and security for CAVs simultaneously. It should be noted that standards in the auto industry, such as ISO 26262, ISO/PAS 21448, and SAE J3061, cannot quantify safety requirements, especially when it comes to the death toll [42].

Table 3 illustrates the projected death rates for automated vehicles [46]. The benchmark refers to the human pilot who is highly trained for safety. In the benchmark, the fatality rate reaches $5 \times 10^{-7}$/hour per vehicle, i.e., one hundred people died every day. If the reliability of CAVs can be quantified, adopting effective CAVs countermeasures can save more than three hundred thousand lives as long as the failure rate is one order of magnitude smaller than that of human pilots.

Based on the in-depth analysis of DHR architecture workflow, some pioneering quantification research directions are discussed as follows and We will explore them in future work.

- Markov reliability models. The scheme of DHR can be modeled by a continuous time Markov chain (CTMC), which can simulate the transition between different states of the system. The final stable state distribution of CTMC can be derived, which reflects the reliability of DHR architecture over time.
- Combined reliability models. In this case, a new perspective from the arbitration cycle is introduced into the quantification of DHR. In the DHR scheme, the arbiter detects inconsistent executors and makes correct judgments within one fixed time threshold. However, the attack outputs during the disjoint arbitration cycle can be interrelated. Mathematical modeling of security and describing the interactions between security and safety is challenging.

Conflict of Interest
The authors declare that they have no conflict of interest.

Data Availability
No data are associated with this article.

Authors’ Contributions
Yufeng Li and Qi Liu designed the model and wrote this paper. Xuehong Chen and Chenghong Cao discussed the recent development and corrected typos in the paper.

Acknowledgements
Thanks to Yuanyuan Liu and the anonymous reviewers for their helpful comments and suggestions.

Funding
This work was supported by the Shanghai Sailing Program (21YF1413800 and 20YF1413700), the National Science Foundation of China (no. 62002213), the Program of Industrial Internet Visualized Asset Management and Operation Technology and Products, Shanghai Science and Technology Innovation Action Plan (No. 21511102502, No. 21511102500), and Henan Science and Technology Major Project (No. 221100240100).
References


Yufeng Li is a professor at the School of Computer Engineering and Science, at Shanghai University, China. His research interests include cybersecurity, broadband information network, and high-speed router core technology.

Qi Liu is a Ph.D. candidate at the School of Computer Engineering and Science, Shanghai University, China. His research interests include the safety and security of connected automated vehicles and privacy protection.

Xuehong Chen is a Senior Professional Engineer of China Industry Control Systems Cyber Emergency Response Team, a member of the China Electrical Engineering Society Energy Internet Committee and China Energy Research Association Smart Power Generation Committee. She has engaged in the research and practice of cybersecurity policies, standards, management, and technologies.

Chenhong Cao is a lecturer at the School of Computer Engineering and Science, at Shanghai University, China. Her research interests mainly focus on the network and system of the Internet of Things, network measurement and security, and wireless sensing.