Systematic Risk Characterisation of Hardware Threats to Automotive Systems

JAMES PICKFORD, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland
RASADHI ATTALE, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland
SIRAJ SHAIKH, Swansea University, Swansea, United Kingdom of Great Britain and Northern Ireland
HOANG NGA NGUYEN, Swansea University, Swansea, United Kingdom of Great Britain and Northern Ireland
LEE HARRISON, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland

The increasing dependence of modern automotive systems on electronics and software poses cybersecurity risks previously not factored into design and engineering of such systems. Attacks on hardware components, communication modules and embedded software – many of which are purposefully designed for automotive control and communications – are the key focus of this paper. We adopt a novel approach to characterise such attacks using Gajski-Kuhn Y-charts to represent attack manipulation across behavioural, structural and physical domains. Our selection of attacks is evidence-driven demonstrating threats that have been demonstrated to be feasible in the real-world. We then risk assess impact of such threats using the recently adopted ISO/SAE 21434 standard for automotive cybersecurity risk assessment, including mitigations for potential adoption. Our work serves to provide unique insights into the complex dynamic of hardware vulnerabilities and how the industry may address system-level security and protection of modern automotive platforms.

CCS Concepts: • **Security and privacy → Embedded systems security**; **Hardware attacks and countermeasures**; **Systems security**.

Additional Key Words and Phrases: Automotive Hardware Security, Automotive Cybersecurity, Threat Analysis and Risk Assessment

1 INTRODUCTION

As technology advances and modern vehicles become more connected and software-driven, the automotive cybersecurity threat landscape becomes more extensive [13]. There is increasing evidence of threats directly manipulating hardware components or characteristics arising out of hardware elements. A modern car is a closely interconnected system, and an attacker inside the outer layers of protection is able to cause untold damage through low-level hardware attacks. Despite the relative increase in difficulty of executing these attacks due to a smaller window of opportunity, the attacks circumvent numerous existing safeguards. Hardware security is therefore of increasing importance due to both the nature of vulnerabilities that emerge from certain technologies or the necessity for effecting controls and mitigation beneath the software layer. Insisting on disrupting deeper down the system stack may be a challenge due to the economic cost, supply chain constraints and various other incompatibilities [47].

Authors' Contact Information: James Pickford, james.pickford@siemens.com, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland; Rasadhi Attale, ras.attale@siemens.com, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland; Siraj Shaikh, s.a.shaikh@swansea.ac.uk, Swansea University, Swansea, United Kingdom of Great Britain and Northern Ireland; Hoang Nga Nguyen, h.n.nguyen@swansea.ac.uk, Swansea University, Swansea, United Kingdom of Great Britain and Northern Ireland; Lee Harrison, lee.harrison@siemens.com, Siemens DISW, Cambridge, United Kingdom of Great Britain and Northern Ireland.

ACM 2833-0528/2024/4-ART
https://doi.org/10.1145/3661315

This paper is motivated by the recent developments in the automotive industry notably the UNECE Regulation 155 [48] provisioning for vehicle cybersecurity and cybersecurity management systems. This regulation mandates a deeper understanding and mitigation of cybersecurity risks and recommends the practice of an international standard ISO/SAE 21434 [17]. This standard is increasingly adopted as a wider practice in cybersecurity engineering for road vehicles throughout the development of their electrical and electronics systems, including their components and interfaces.

The complexities of some of the attacks on automotive hardware present a complex picture, given the diversity of attack vectors, the nature of vulnerabilities exploited and potential mitigations. Our work attempts to systematically characterise a selection of such attacks within the threat analysis and risk assessment framework offered by ISO21434. This framework enables us to comprehensively enumerate the attacks' possible impacts and objectively evaluate their corresponding ratings on relevant aspects including safety, financial, operational and privacy. Additionally, their feasibility evaluation is also standardised according to different aspects including elapsed time, specialist expertise, knowledge of the item (or component), window opportunity, and equipment. As such, this paper contributes towards a systematic and comprehensive coverage of a range of attacks targeting vulnerabilities arising out of limitations due to hardware components and implementations, and ultimately offering a system-level view of risks and mitigations in automotive systems.

1.1 Rest of this paper

The rest of this paper is organised as follows. Section 2 provides a background to the nature of cybersecurity threats targeting automotive systems, with a discussion on notable contributions in the relevant literature. Section 3 describes the methodology adopted in this paper in terms of hardware domains exploited and the risk assessment and mitigation framework. Section 4 presents the main contribution of this paper delving into the characterisation of each threat alongside typical mitigations. Finally, Section 5 concludes the paper with a discussion reflecting on some of the key insights emerging from this work.

2 RELATED WORK

Threats to vehicles in the modern era come from a variety of vectors, having exceptional variety in the range and method of access. There have been various efforts to survey existing threats and attacks in the automotive domain. Often, these focus on examining the car as a whole - papers such as those by Checkoway [10], Al-Sabaawi [3] and Khan [24] model the car as a connected device, and focus much of their efforts onto that connectivity. Others focus on a single facet of a car, such as a specific ECU or feature. This could be the sensors of a vehicle [14], [46] specific peripherals on the OBD-II port [12], or the LiDAR-based aspects of an autonomous driving system [8]). Moschos in particular looked into the possibility of remotely-activated hardware trojans that may increase the attack surface to expedite more hardware attacks listed [32]. Similarly, Coppola [13] highlighted potential threats to connected cars that can originate from vehicle elements such as ECUs, associated mobile/embedded apps, OBD-II ports, CD-players, USB port, and CAN-bus/wireless networks; however they do not offer any evidence of actual hardware attacks. However, none have focused solely on attacks directly exploiting the hardware of the vehicle. Despite the relative increase in difficulty of executing these attacks due to a smaller window of opportunity, the attacks circumvent numerous existing safeguards. A modern car is a closely interconnected system, and an attacker inside the outer layers of protection is able to cause untold damage through these lower-level attacks. To the best of our knowledge, our paper is the first to solely focus on hardware-based attacks on automotive systems, where vulnerabilities in hardware design or implementation are a key part of the attack vector and where the attacks have been demonstrated successfully.

From our point of view, the threats in the automotive domain can be categorised into three main categories based on the broad category of systems they are targeting: Threats to Control Systems, Threats to Vehicle Sensors,

and Threats to V2X Communication. Each of them can be divided further according to specific technologies as potential vectors.

2.1 Threats to Control Systems

Attacks on automotive control systems involve compromising a device or wider network within the vehicle, potentially allowing for lateral movement throughout the vehicle. This category encompasses technologies that control the vehicle, containing the in-vehicle networks and physical access ports. These attack vectors typically require physical access to use, but this is not always the case.

2.1.1 OBD-II. Attackers target OBD-II, the eponymous port on the vehicle, as a method of accessing its interior network – and thus, the critical systems within. As this port is for maintenance and onboard diagnostics, it has no network connection – meaning that the attacker would have to have physical access to this port to use this vector.

One method of accessing the system through the port is with malicious OBD-II devices, whether reverse-engineered or otherwise compromised. Checkoway et al. [10] demonstrated a compromised OBD-II device – namely, a PassThru device – which was used to inject arbitrary code into the internal CAN network of the car. The Pass-Thru device was standardized by SAE J-2534 requirements in 2004 to use specific parameters and accept powertrain reprogramming, thus making the device an attractive vector for threat actors. This attack may potentially circumvent the physical access requirement if the diagnostic computer using the compromised device was connected to the internet and thus compromised remotely.

The port is also vulnerable to man-in-the-middle attacks – a study performed by Christensen and Dannberg demonstrated the use of the aforementioned attack on an OBD-II "dongle" to successfully intercept all data passing through the device and additionally inject arbitrary CAN packets into the network. Further in the study, this led to the potential for DoS (Denial of Service) attacks and arbitrary code execution [12].

2.1.2 CAN. The CAN network is the connecting fabric of the vehicle's components, and as such many of the attacks focused here will concentrate on the potential for lateral movement within the vehicle once a foothold is gained through a compromised node. Fröschle et al. [15] presented a study that analysed the capabilities of the CAN-based attacker which presented some possibilities of what a threat actor could do when inside this network, including but not limited to silencing nodes on the network, injecting arbitrary packets, and impersonating existing nodes. This gives the attacker full control over most aspects of the car - leading to potentially hazardous situations such as adjusting power steering angles.

Malicious hardware devices are also a vector here, potentially bypassing the need for access to the OBD-II port by being wired directly into the network. The introduction of a commercially available mileage tampering device compromises the authenticity of the CAN network by blocking a number of packets with a certain header passing through it, decreasing the logged mileage by an adjustable amount [38].

2.1.3 ECUs (Electronic Control Units). These are the most common of the computers found in vehicles, controlling everything from monitoring systems to in-car entertainment. They are also a large security weakness. There are typically 15-100 in a vehicle and the majority seem to be "surprisingly insecure" [42]. If an ECU is compromised, the unencrypted nature of in-car networks is easy to exploit, as expanded upon in the previous section. Roufa et al. [39] utilized the ECU within the Tyre Pressure Monitoring System (TPMS) to extract the data within a vehicle's network – made possible by the lack of verification of input values of the TPMS and the fixed sensor IDs making it easy to identify key nodes in the vehicle.

Reverse engineering of ECUs is also a viable vector – readily available off-chip technology is being used on physical ECUs in order to strip the PCB and extract data directly, expediting the process of reverse engineering the chip to provide easier access into the vehicle network [38].

2.2 Threats to Vehicle Sensors

Attacks targeting vehicle sensors are often targeted towards producing adverse behaviour in safety systems rather than data exfiltration, owing to the nature of the systems that these sensors influence. However, attacks are not limited to this effect, as demonstrated by research later in this section. These sensors often are a key part of autonomous driving systems, and as such many attacks listed here will target that system itself using the sensor as an entry point.

2.2.1 LiDaR. This sensor is used for autonomous driving and is exploitable via the Machine-learning-based algorithms used for autonomous driving. Cao et al. [8] present an attack vector using 3D-printed "adversarial objects" that resemble danger scenarios to the ML system. This misleads driving systems into thinking an object is dangerous, thus creating a hazardous situation when a response is provoked by the system.

2.2.2 Radar. Like most sensors, Radar has two main methods of attack – spoofing and jamming. Yan et al. [52] demonstrated both of these methods on a facsimile of the Tesla Model S' radar system, utilizing a fixed 45MHz waveform to jam the radar and prevent detection of a simulated car. Spoofing proved to be less effective, yet some success was had in the form of observable distance changes. Komissarov and Wool [27] had more success with spoofing – managing to manipulate both the range and velocity of an object to simulate a full range of movement. This has the potential to provoke an adverse response in the vehicle with regards to safety systems, for example: preventing or triggering emergency braking systems.

2.2.3 GPS. This sensor is, again, vulnerable to both spoofing and jamming – with GPS spoofing being one of the most applied attacks in the automotive domain [35]. GPS spoofing can be broken down into 2 main categories: Location spoofing, whereby a vehicle can be lured to a different location by sending false information.

GPS time spoofing can be achieved by overpowering the received GPS signal with new, stronger signals. These signals would be signed from a date some period in the future (e.g. 1 day). By using these signals, an attacker can control data sent to the unit, allowing the vehicle to be tracked [5].

GPS Jamming is one of the easiest methods to attack a vehicle. The method utilises a generator of large amounts of signal noise in the operating frequency of the GPS unit to overwhelm incoming transmissions and thereby prevent the vehicle from being located [14]. This may lead to the vehicle ending up in unwanted and potentially dangerous locations if the driver is unfamiliar with the locale.

2.2.4 Camera. A common attack on this sensor is Camera Blinding. It occurs when the camera is not able to turn down the auto-exposure anymore [46]. This prevents the camera from proper operation by being unable to recognize or process images properly, in much the same way as a person's eye cannot see when exposed to a much brighter light than the surrounding light level.

Adversarial images can also be employed in a similar fashion as with LiDAR, whereby an image designed to provoke a dangerous reaction is introduced to the camera's line of sight. This is effective against deep learning models used for autonomous driving, with potential applications for other safety systems. Such attacks differ from LiDAR-based attacks, however, in that the images are designed to be imperceptible to humans [36]. The camera, however, recognizes these patterns and applies a response targeted by the attack, e.g. the braking system.

2.3 Threats to V2X Communication

V2X communication is one of the defining features of newer vehicles and is one of the largest attack vectors for any automobile. This vector eliminates one of the most common limitations of automobile cyberattacks, which is proximity to the vehicle. Modern connectivity allows for both short and long-range attacks on vehicles previously impossible before the introduction of this technology. This is also the most active area of research for automotive cyberattacks [26].

2.3.1 Mobile Apps. Mobile Apps are a large part of smart automation, with modern applications such as the Tesla app allowing users to view information such as current firmware version, current drive mode, VIN, and odometer. It also provides functionality such as locking and unlocking the vehicle and opening the trunk. This provides an incredibly attractive vector to adversaries – and for good reason. Chatzoglou et al. [9] performed an analysis of the top official Car Management apps for Android and found that at least 87% of the apps were found to be exposed to 6 of the 11 common weaknesses listed in the CWE (Common Weakness Enumeration) database. Additionally, at least 80% of the apps included obsolete cryptographic algorithms. Roughly 87% of these apps also had more than one issue in their manifest files.

This provides an opportunity for several attacks – Renganathan et al.demonstrated a "valet attack", utilizing the Bluetooth connection from a mobile phone to the infotainment system to extract data from that mobile. After the Bluetooth connection is severed, information from the mobile device remains on the infotainment unit and can be extracted by someone with physical access to the car – e.g. a valet [37].

Another proposed attack structure involves using a malicious application on a user's mobile in order to send commands over Bluetooth, potentially chaining into a buffer overflow to cripple vehicles via commands sent over the in-vehicle network [18]. This attack is potentially scalable over a fleet of similar vehicles provided enough cellular devices are compromised, and the aforementioned study proposes a large-scale attack on infrastructure potentially crippling transportation using this method.

2.3.2 OTA (Over-The-Air) Updates. This software update technique is a necessity in the modern environment where large updates are needed to be deployed regularly. However, this technology itself is vulnerable. Nilsson et al. [33] suggested an impersonation of the portal the vehicle uses to retrieve the update itself, allowing for the injection of malware remotely. They also posit that an intrusion into a weakly secured portal in order to exfiltrate data on the vehicle or compromise the integrity of the portal to compromise all updates passing through it.

Karthik et al. [23] suggest further methods of attack in their publication, as follows: Denying access to updates through a compromised portal in order to continue exploiting other software vulnerabilities, rolling back existing updates in order to exploit previous vulnerabilities, sending the update ECU an infinite amount of data to induce a crash, and controlling admissible software to cherry-pick installed updates.

2.3.3 Infotainment Systems / Head Units. These components need internet access to perform their function, yet this makes them vulnerable to attacks – Miller and Valasek's remote exploitation of a Jeep Cherokee [31] utilized an exposed cellular network port in the car to compromise the infotainment system, leading to injection of malicious CAN packets to the in-vehicle network, and jailbreaking of the head unit.

Checkoway et al. [10] also propose reverse engineering the CD player component of the head unit, using a standard ISO-9960 formatted CD to upload arbitrary data to the infotainment unit. The media player in the same device may also become compromised, allowing for arbitrary CAN packets to be sent over the network through the use of modified WMA audio files. Finally, they suggest the reverse engineering of the Bluetooth interface in the head unit to allow for code injection via a stack-based buffer overflow.

2.3.4 Automotive Key. The automotive key is a necessary method of authentication to enter a vehicle, and with the rise of keyless entry systems (and remote start) this has become the prime vector to use in order to steal the vehicle itself.

Aerts et al. [1] demonstrated a practical attack on the KeeLoq algorithm, used to provide encryption for keyless entry systems, in 2018. They used a side-channel attack on power traces throughout the ECU to manage this system in order to extract the manufacturer key and access the vehicle.

More recently, an attack on the keyless entry system of the Tesla Model X bypassed the cryptographic protocols of the keyless entry system entirely. This allowed the attackers to start any Model X, anywhere, that was running the specified version of keyless entry protocol [50].

2.3.5 VANETs (Vehicular Ad Hoc Networks). VANETs are networks that spring up between nearby vehicles to share information, such as traffic information [45]. Attackers may use these networks to perform a variety of attacks by utilizing the unique peer-to-peer architecture.

A Sybil attack uses several false identities to create several fake vehicles, or fake nodes, on a VANET. These can artificially damage a roadway and the decision-making of a vehicle by flooding the target with false information regarding surroundings, such as current traffic or active brake lights [3]. Another attack in the same vein was proposed by Al-sabaawi et al. [3], called a Masquerade attack; a vehicle conceals its identity and appears to be a legal node in the VANET. Strangers can construct more subtle attacks such as false messages or injected malware through these.

Falsified GPS information broadcasted over VANETs can also be used in a black hole attack: an attack whereby a rogue node broadcasts that they have the shortest route to a given destination, then prevents the requesting vehicle from receiving any destination information. This may eventually crash the VANET [21].

3 METHODOLOGY

In this paper, we investigate a selection of hardware attacks, representative of some of the key threats posed to automotive platforms. For each attack, we (i) break down its full attack vector, (ii) evaluate its associated risk and (iii) propose potential mitigation solutions. In the sequel, we summarise each of these three steps in more detail.

3.1 Attack vector analysis

In the first step, the attack vector is analysed based on the Gajski-Kuhn Y-chart [22]. The Y-chart, depicted

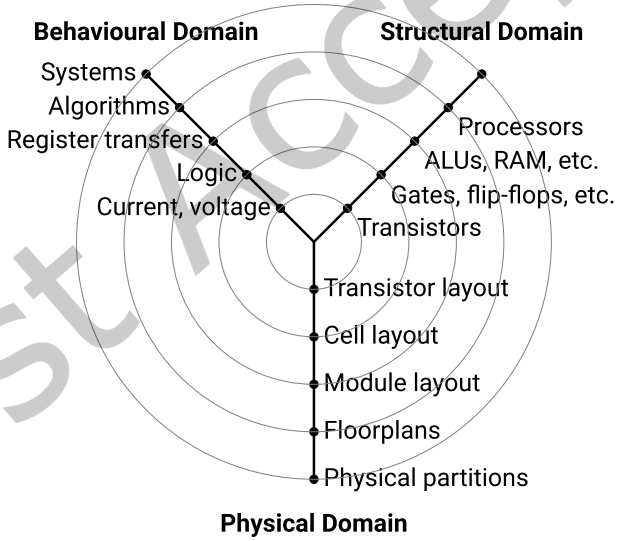

Fig. 1. The Gajski-Kuhn Y-chart illustrates the hardware design activities from high to low abstract levels (outer to inner circles) in 3 different domains: behavioural, structural and physical.

in Figure 1, illustrates a general approach to designing hardware where activities are grouped into different abstraction levels (circles) and domains (axes). Design activities start from the highest abstraction level where designers are given a specification describing what the system will do. It is broken down into how this can be realised in terms of algorithms. Then, design activities move to the structural domain to specify components to

run this algorithm and to the physical domain to arrange these components. The description of these components then becomes the specification for the behaviour domain in the next lower abstraction level where the design activities repeat themselves, but, at this lower abstraction level.

In this paper, we utilise the Gajski-Kuhn Y-charts to provide a brief overview of exactly which levels of abstraction the attacks target in relation to hardware. They are sourced from the most recent version of the cyber security body of knowledge [49]. The three domains relate to behaviour, structure and the physical chip itself that is targeted. These can be mapped to provide an image of how abstracted the attack itself is - for example, reverse engineering of the chip itself would have a small footprint on the Y chart, with a low-level attack - but interference with the chip's communication protocols would target a higher level of abstraction, and thus have a higher footprint. We rank these charts based on the highest level of abstracted detail that the attacks reach.

3.2 ISO/SAE 21434 Standard Risk Assessment

To understand the feasibility and the impact of these attacks, we use a Threat Analysis and Risk Assessment (TARA) process as described and explained in ISO/SAE 21434 standards [17]. This standard is becoming a common practice in the automotive industry where TARA activities can be mapped onto the engineering process [44]. This TARA process includes two qualitative classifications: (i) a class-based classification for attack impact and (ii) a numerical measure of attack feasibility. Detailed guidelines for this TARA process can be found [17] (appendices F and G). In the following, we briefly recall the numerical ratings for attack feasibility and the categorical ratings for impact.

3.2.1 Impact rating. The impact of an attack is determined in four different aspects: safety, finance, operation and privacy according to ISO/SAE 21434. The overall impact of the attack is the highest impact level according to any of these aspects. ISO/SAE 21434 classifies levels of safety consequences in terms of human casualties using safety impact ratings from IS026262-3:3018. A severe rating represents deadly injuries with uncertain survival, while a major rating indicates deadly injuries with possible survival. A moderate rating signifies light injuries and a negligible rating implies no injuries. They are summarised in Table 1.

Impact rating	**Criteria for safety impact rating**
Severe	S3: Life-threatening injuries (survival uncertain), fatal injuries
Major	S2: Severe and life-threatening injuries (survival probable)
Moderate	S1: Light and moderate injuries
Negligible	S0: No injuries

Table 1. Safety Impact Ratings taken from IS26262-3:3018, as specified in ISO/SAE 21434, are highly relevant in this context given the automotive systems are safety-critical. Such definitions are adequately given in the severity of the nature of injury to passengers.

Finance impact ratings determine the level of financial consequences in the presence of an attack. The severe rating signifies high financial damage that the stakeholder may not be able to overcome. Major rating represents significant financial damage that the stakeholder will be able to overcome. A moderate rating means that it will cause inconvenient results, but the stakeholder will be able to overcome it with limited sources. A negligible rating implies no financial damage, and the stakeholder will not take any action. In this paper, we specify these ratings in more detail. In particular, we classify severe financial damage from a successful attack that leads to legal action or severe medical bills levelled at the end road user, which may not be covered by insurance. Major would be financial damages roughly equivalent to the loss of the vehicle. Moderate impact would be equivalent to major damage to the car, such as damage to bodywork or replacement of interior components. These ratings are summarised in Table 2.

Impact rating	Criteria for financial impact rating
Severe	Financial consequences are catastrophic for the affected road user to overcome.
Major	Financial consequences are substantial. However, the affected road user can overcome them.
Moderate	Financial consequences are inconvenient for the affected road user. He/she can overcome them easily.
Negligible	Financial consequences are negligible or irrelevant to the affected road user.

Table 2. Financial impact is notoriously difficult to characterise given that such impact may arise out of a number of liabilities due to assets inside of the vehicle and outside in terms of other road users and infrastructure. Actual monetary values assigned to these rankings are expected to emerge as the application of ratings are assessed for against actual incidents.

Operational impact ratings are related to the levels of operational consequences. A severe rating indicates that the vehicle becomes non-operational. A major rating signifies the loss of vehicle function. For example, a major impact would be the disabling or compromisation of a key vehicle function such as the disabling of the ABS. A moderate rating implies partial dysfunction or performance loss of the vehicle. For example, a moderate impact would be the degradation of important functions, such as excessive latency within the head unit of the vehicle or the inability to move the electric windows. A negligible rating means there is no effect on the vehicle's function or performance from the damage. Table 3 summarises different ratings for operational impact.

Impact rating	Criteria for operational impact rating
Severe	Operational damage leads to the loss or impairment of a core vehicle function. EXAMPLE 1 Vehicle not working or showing unexpected behaviour of core functions such as enabling limp home mode or autonomous driving to an unintended location.
Major	Operational damage leads to the loss or impairment of an important vehicle function. EXAMPLE 2 Significant annoyance of the driver.
Moderate	Operational damage leads to partial degradation of a vehicle's function. EXAMPLE 3 User satisfaction is negatively affected.
Negligible	Operational damage leads to no impairment or non-perceivable impairment of a vehicle function.

Table 3. Rating operational impact is perhaps relatively straightforward as this classifies impact in terms of the functional aspects of the vehicle where severity could be attributed to loss of critical and non-critical features available to the vehicle user.

Finally, the **privacy impact rating criteria** classify the level of impact on the road user and the sensitivity of the information involved. The severe rating signifies an irreversible impact on the road user, with highly sensitive information that can easily be linked to the Personally Identifiable Information (PII) principal (see ISO29100:2011). A major rating indicates a serious effect on the road user, with highly sensitive information either difficult to link to PII or sensitive and easy to link to PII. A moderate rating signifies inconvenience caused to the road user, with information that is sensitive but challenging to link to PII or non-sensitive but easy to link to PII. A negligible rating implies no effect on the road user, with information that is not sensitive and difficult to link to PII. These impact ratings are summarised in Table 4.

3.2.2 Attack Feasibility Rating. The feasibility of an attack indicates the ease or difficulty of carrying out an attack on four distinct levels. A high feasibility level suggests that the attack path is easy to accomplish. A

Impact rating	Criteria for privacy impact rating
Severe	The impact of privacy damage is significant and not recoverable for the affected road user. EXAMPLE 1, disclosure of highly sensitive information which is directly linked to and identifies a PII principle leads to identity fraud and/or theft.
Major	The impact of privacy damage is serious but potentially recoverable for the affected road user. EXAMPLE 2, disclosure of highly sensitive information (but not directly to link to and identify a PII principal) such as the leak of username and password.
Moderate	The impact of privacy damage is inconvenient to the affected road user. EXAMPLE 3, disclosure of sensitive information (but not directly link to and identify a PII principle) such as the disclosure of phone number or email address.
Negligible	The impact of privacy damage is negligible and not relevant to the affected road user. EXAMPLE 4, disclosure of insensitive information that is not directly linked to and identify a PII principle.

Table 4. Privacy impact is a relatively new concept for automotive systems, arising out of concerns around exposure of personal data and behavioural monitoring. The criteria for rating varying levels of such impact is still to be tested in practical assessments. ISO/SAE 21434 makes reference to the Personally Identifiable Information (PII) categories as per ISO29100:2011.

medium feasibility level implies that the attack path is feasible and commonly encountered. A low feasibility level indicates that the attack path is feasible to some extent. Finally, a very low feasibility level suggests that it is highly challenging, if not nearly impossible, to accomplish the attack path.

In our work, the attack feasibility is determined by the attack potential-based rating approach introduced in ISO18045:2022. This is one of the three approaches suggested in ISO21434. In this approach, one has to take into account different aspects of the attack including elapsed time, specialist expertise, knowledge of the item (or component), window opportunity, and equipment. The ratings for each of these aspects are summarised in Table 5.

Elapsed Time		**Specialist Expertise**		**Component knowledge**		**Opportunity Window**		**Equipment**	
Enumerate	Value	Enumerate	Value	Enumerate	Value	Enumerate	Value	Enumerate	Value
<1 week	0	Layman	0	Public	0	Unlimited	0	Standard	0
<1 month	1	Proficient	3	Restricted	3	Easy	1	Specialized	4
<6 months	4	Expert	6	Confidential	7	Moderate	4	Bespoke	7
≤ 3 years	10	Multiple Experts	8	Strictly Confidential	11	Difficult	10	Multiple Bespoke	9
> 3 years	19								

Table 5. Rating attack feasibility according to different aspects of an attack is a substantial task given the inherent nature. Such feasibility is also likely to be affected as threat actors and the resources available to them evolve, beyond just automation and proliferation of tools and techniques.

The rating scores in Table 5 reflect the relative severity of the potential for an attack to happen. The first column encompasses the time since an attack was published and reflects the time available to develop an exploit or mitigations. The Expert Knowledge reflects the degree of technical knowledge needed to develop the exploit in the first place, whilst the technical knowledge needed about the specific components exploited, such as memory structure and clock skew, are represented by the Component knowledge column. The Opportunity Window

represents the time needed to fully complete the attack, and the Equipment column represents the cost and difficulty of acquiring the non-standard equipment needed to perform the attack. Higher scores here represent a lower likelihood of the attack being successfully performed. Once the attack feasibility ratings for each of these aspects of the attack are determined, its overall feasibility rating is identified based on their sum. The map between the sum to the overall attack feasibility rating is depicted in Table 6.

Attack Feasibility Rating	Values
High	0-13
Medium	14-19
Low	20-24
Very Low	25+

Table 6. This rating supports the accumulative value calculation from Table 5 to help reason and compare attack feasibility across a range of diverse attack methods.

3.3 Risk mitigations

Once the impact and the feasibility ratings of an attack are determined, ISO 21434 suggests four options for risk treatment.

Risk reduction: This involves mitigation of risk through some reconfiguration or a control mechanism introduced purposefully. Such mitigation may not entirely mitigate risk, but only does so partially;
Risk avoidance: This involves overcoming the risk by removing a component or a process that is the main cause of the risk. Such avoidance would typically mean the loss of a component or a feature as a result of the avoidance;
Risk sharing: This may involve proposing some technical or operational design intervention that results in the sharing of risk across a number of assets (components) or stakeholders owning the risk; and
Risk acceptance: Typical risk assessment would lead to some residual risk, which may be deemed to be acceptable. As such, a cybersecurity case (corresponding to a safety case) may be presented to argue for retaining such a risk. Such a case may become part of a formal claim subject to approval by a regulatory body for further evaluation and audit.

In this paper, we will discuss several techniques and methods to mitigate each of the attacks as technical solutions without sacrificing the system's functionality.

4 HARDWARE ATTACKS AND RISK MITIGATION

We have shortlisted a selection of ten threats based on the following criteria:

- The threats target some hardware component, either through direct physical interaction, or remotely through a wireless interface that ultimately exploits a vulnerability present in the implementation of the hardware layer. As such, this excludes threats that take advantage of the exploitation of vulnerabilities due to software-only implementation of a feature;
- The automotive features and components targeted by such threats were mature implementations widely available in the industry, as opposed to concepts and futuristic designs or prototypes typically available in research literature alone;
- The threats have been demonstrated successfully, in enough detail and published through a notable technical or scientifically peer-reviewed source. This is important to establish credibility both on the method and

the team presenting the threat evidence. This also reassures on responsible disclosure of vulnerabilities exploited.

The rest of this section is organised as ten subsections each dedicated to describing the threat, an assessment of the typical nature of the resulting risk, and known mitigations that serve to either reduce the risk from a threat either entirely or partially.

4.1 Reverse Engineering key fob

4.1.1 Attack Description and Asset Identification. A major component of the physical security of a vehicle is the key fob. By reverse engineering this component, an attacker may gain control of a vehicle and its contents. Wouters [50] propose using reverse engineering of the key fob and controller system to duplicate a key and thereby steal a vehicle. Therefore, the asset compromised in the attack is the key fob. Reverse engineering of the Keyless entry system allows for compromisation of the Remote Keyless Entry system (RKE) of the car, allowing it to be remotely started. The targeted ECU is the Texas Instruments Bluetooth Low Emission (BLE) SoC within the vehicle key fob, Body Control Module (BCM). The attack uses modified Original Equipment Manufacturer (OEM) hardware to expedite the process of the attack, consisting of a modified BCM and manufacturer standard key fob. The attack flow proceeds as follows: the attacker's BCM broadcasts a signal over BLE in order to wake up the legitimate key fob for the vehicle, addressing the broadcast via the Vehicle Identification Number (VIN) for the targeted vehicle (which can be read off of the windshield of the vehicle). Then a connection is made to the legitimate fob via BLE and a malicious firmware update is sent. This update is legitimised by using the Tesla Toolbox service for firmware, which is freely available. The update then disables the block list within the ECU in order to permit the transmission of a vehicle unlock token over BLE. The attack vectors of this attack are summarised in Figure 2.

Removal of this block list allows for the adversary to request this token from the key, which is then transmitted over to the second "dummy" key in the attacker's possession. This essentially duplicates the legitimate key. Finally, the second key must be paired by connecting to the OBD-II interface within the Tesla. This finally allows the adversary to duplicate the key fully, and take control of the vehicle.

4.1.2 Attack Feasibility and Impact Assessment. This attack would usually happen when the vehicle owner is in a public place, with their car parked somewhere accessible to the public. The attacker would carry the malicious hardware on their person, and close into the vehicle owner in order to get within Bluetooth range of their legitimate key fob. This means that the window of opportunity is variable based on how often the car is driven, where the vehicle is stored, and where the key is stored. However, we believe that for the purposes of this study, the car will be parked in a publically accessible place such as a car park regularly, increasing the window of opportunity greatly. Specialist expertise is needed to create the hardware for the attack, but once produced, the hardware is usable with a much lesser amount of expertise and will work on any unpatched key fob with the vulnerability. A great deal of component knowledge of the key fob is needed to make the malicious hardware, but not to perform the attack itself. Finally, equipment for the attack is difficult to acquire but can be sourced via manufacturer channels. The attack feasibility level of this attack is summarised in Table 7.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
0	3	3	1	7	14	Medium

Table 7. The attack feasibility is rated as medium, largely down to the bespoke nature of the hardware needed to reverse engineer. Most other elements of this attack are typical to such hardware attacks in terms of access and knowledge required.

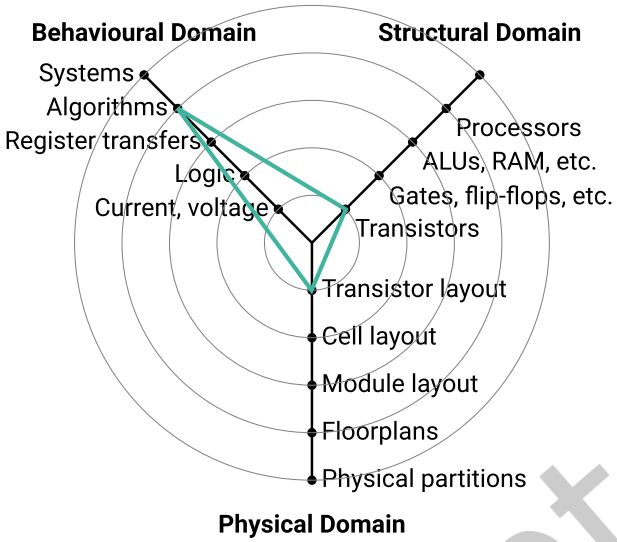

Fig. 2. Attack Hardware Classification: Reverse Engineering key fob. The attack targets the transistors on the chip and the transistor layout due to the necessity for the custom hardware manufactured for the attack. Behavioral targeting is limited to the algorithms needed to pair a new key fob to the vehicle, including the power traces and the function of the Bluetooth link.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Moderate	Major	Moderate	Major	Major

Table 8. Theft of vehicle translates to both financial impact, in terms of the value of the vehicle, and privacy impact, in terms of access to personal data and contents in the vehicle. Overall, this is assessed to be major given that, over and beyond the theft, the vehicle may also be exploited for other crimes.

Safety impact is justified as moderate here as losing access to your vehicle may cause injury or inconvenience due to not being able to return from a destination that you took the car to, such as a remote location. The loss of the key fob integrity often leads to the loss of the car, incurring significant financial harm to the owner. It also leaves the operation of the vehicle significantly impaired if the vehicle is stolen, and unsanctioned access to the vehicle may lead to contents stolen as well, as well as the loss of the information on the head unit or linked devices. The impact ratings of this attack is summarised in Table 8.

4.1.3 Risk Mitigation. In risk mitigation the asset that is focused on is the key fob. Reverse engineering of the key fob was possible as the key fob did not include any capabilities to verify that the firmware was legitimate. As Wouters et al. [50] proposed by including a random firmware signature in a legitimate firmware update and by verifying this signature by the receiving key fob such malicious firmware updates can be blocked. In this particular attack scenario, Secure Element is used to verify the SHA-1 hash digest and random firmware signature via a public key. However, this mitigation only partially mitigates as it is prone to an SHA-1 collision attack. Wouter et.al commented that the secure public-key and symmetric-key primitives implemented in a Common Criteria certified Secure Element is insufficient due to this reason. An adversary can generate a firmware update with a SHA-1 digest that collides with the SHA-1 digest of the signed firmware.SHA-1 collisions can be detected

and blocked using a detection algorithm. The methodology proposed by Al-Odat et al. [2] looks at compression calculations which are capable of detecting SHA-1 collisions for two blocks.

The method starts with initial blocks (IHV), where IHV_0 is the input to block 1 and an 80-step compression function is used to calculate IHV_1. IHV_1 is the input to the second block to obtain IHV_2. For messages that include a collision attack backward computation is $IHV_2' = \delta IHV + IHV_2$ where δIHV is a disturbance vector. This backward computation will result in IHV_0' and IHV_0 being equal. If both IHV_0' and IHV_0 are equal then the algorithm returns true, and a collision is detected. False if not.

Verifying the firmware with an encrypted signature that is strengthened against collision attacks mitigates the risk of a key fob attack.

4.2 Electromagnetic Fault Injection (EMFI)

4.2.1 Attack Description and Asset Identification. Components of a vehicle are often trusted that they are secure when booted, and would be compromised later. However, if an attacker manages to compromise the ECU completely down to the source code, the assumption of trust here can be leveraged to perform a large suite of attacks not feasible otherwise. O'Flynn et al. [34] seek to leverage this privilege by compromising the boot sequence of an ECU using a physical fault injection technique. This allows for full access into the ECU, and control over the entire source code itself.

The attacker begins the attack by identifying where the flash memory is stored for the Boot Assist Module (BAM). Therefore, the asset compromised in the attack is the flash memory.This will have an access password stored in a specific location that controls legitimate changes to the memory, and thus control over all functions of the ECU. They then access the BootCFG1 pin in order to force the ECU into the bootloader. A computer is connected to the Joint Test Access Group port (JTAG) port, with a program that will try to edit FLASH memory with arbitrary code. As the ECU boots, the attacker sends an access password via CAN. This could either be the manufacturer-issued public password, in which case no exploit is needed, or an arbitrary password. To get the BAM to accept the password and allow the attacker to access FLASH, a fault in the control flow is injected via an EMFI device. This induces a voltage in the ECU processor which simulates an electrical fault in a specific area - with specific timing, a return bit can be 'flipped', from a negative (0) to a positive (1). The control flow therefore accepts an incorrect password and echoes the correct password back. This password can then be used to access the memory, giving full control of all functionality of the ECU. The attack vectors of this attack are summarised in Figure 3.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
1	6	7	10	7	31	Very Low

Table 9. The feasibility of the attack is judged to be Very Low, as the complexity of the procedure and its low likelihood of success, coupled with the tiny window of opportunity, means that the attack is unlikely to happen outside of laboratory conditions.

4.2.2 Attack Feasibility and Impact Assessment. This attack would usually happen when the vehicle was parked for a significant length of time, or in service. This, along with having to access hidden components within the vehicle and the large amount of specialist equipment required, results in a very small window of opportunity. The specialist expertise required is also very steep and requires specialist equipment that is hard and expensive

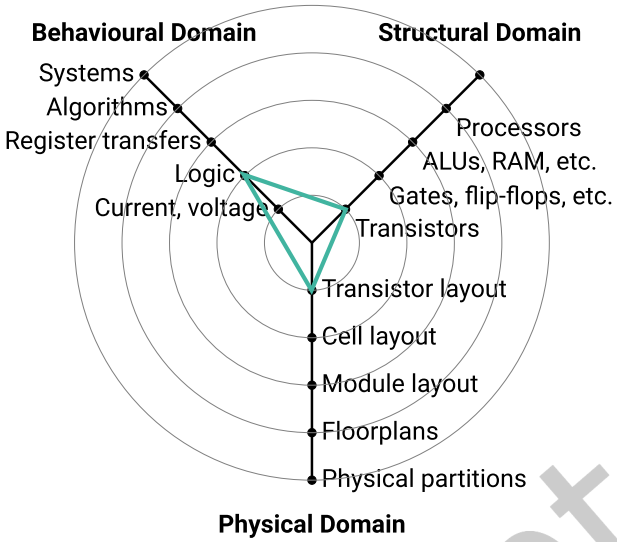

Fig. 3. Attack Hardware Classification: Electromagnetic Fault Injection. The attack targets the transistors on the chip, including exact transistor locations for the boot sequence. The attack targets the behaviour of the logic, namely the booting of the chip and the logical flow of verification. Finally, exact knowledge of transistor layout is needed to properly position the fault injection device.

to acquire. Detailed component knowledge is also required, down to specific memory locations within chips. The attack feasibility level of this attack is summarised in Table 9.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Severe	Major	Severe	Severe	Severe

Table 10. The attack impact is classified as severe, as complete compromisation of the ECU may is expected to enable a foothold to every type of impact here, depending on the further goals of the attacker. Financial and privacy impact are affected thanks to compromisation of the bootloader password, leading to extraction of valuable manufacturer or PII information, whilst safety and operation are compromised thanks to the potential overhaul of a critical vehicle ECU.

If the attack is successful, the ECU targeted is completely compromised - if this relates to a significant vehicle function, this has a high probability of causing severe injuries under certain conditions. Operations of the vehicle could also be disabled, compromised or significantly endangered. All data passing through the vehicle can also be compromised, including providing a potential to chain attacks into connected devices. Finally, the compromisation of this ECU may lead to the loss of the vehicle and the associated financial harm from that. The impact ratings of this attack are summarised in Table 10.

4.2.3 Risk Mitigation. In this attack, a fault injection such as voltage fault injection (VFI) is used to bypass the security measures and compromise the flash memory. In risk mitigation the asset that is focused on is the flash memory. As Boas et al. [6] proposed a Brown-out detection (BOD) circuitry can be used to monitor supply voltage (V_{DD}) drops and reset the system. A BOD circuitry's main purpose is to power down the device when the supply voltage drops below a threshold but it also detects VFI-caused voltage drops. BOD can prevent data

corruption and malfunction as when the V_{DD} falls below the minimum voltage required for data retention, the BOD circuit generates a reset signal. When a fault injection attack is triggered, this reset will prevent the attack from succeeding.

As Marvin et al. [41] discovered if the BOD is a sampling frequency-based detection system by creating μ-glitches the detection can be bypassed. This requires splitting a single wider VFI into several narrow VFIs having the same effect on the target. Marvin et al. [41] also realised that the success rate of a μ-glitch VFI attack is significantly lower as the success rate decreases with each extra glitch.

Based on this a BOD circuitry will fully mitigate against any VFI attacks. Fault injection-based flash memory attacks can be performed by other methods that the BOD does not mitigate against. Therefore this control partially mitigates the risk of fault injection attacks.

4.3 Update Tampering

4.3.1 Attack Description and Asset Identification. Furthering the theme of lower-level compromisations of ECUS offering greater privileges, a malicious Powertrain update of a pass-thru vehicle service device allows for remote exploitation of the device. Checkoway et al. [10] utilise this to push a malicious firmware update to an ECU when the vehicle is in service. This will allow full control of the functionality of an ECU. Therefore, the asset compromised in the attack is the ECU.

All vehicles sold in the US since 2004 are required to support the Pass-Thru standard (SAE J2534) - an interface that standardises interfacing with the On-Board Diagnostics port of a vehicle. The standard provides a Windows API running on a network, and a physical component that plugs directly into the On-Board Diagnostics (OBD-II) port. This Windows API is able to be compromised remotely by an attacker, and through this, the CAN network itself can be compromised. An attacker can compromise the physical Pass-Thru device by exploiting the underlying Linux distribution via shell injection. From here, the attacker can craft a malicious update for the ECU to be targeted, in this case, the telematics unit. By compromising this physical device, hardware exploitations can be performed fully remotely. Moreover, since the physical Pass-Thru device has enough power to perform the attack itself, this attack is fully wormable (self-spreading), allowing for an extremely large attack impact on multiple vehicles in multiple locations. The attack vectors of this attack are summarised in Figure 4.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
19	3	3	1	0	26	Very Low

Table 11. Attack Feasibility here is classified as very low. Whilst the exploit itself is wormable and devastating if it works, and barrier to entry is low thanks to lack of specialist equipment and remote activation, there has been sufficient time to patch this particular exploit via updates to the pass-thru device.

4.3.2 Attack Feasibility and Impact Assessment. This attack would usually happen when the vehicle was in service. If the adversary had access to the garage where the vehicle was stored, this allowed for a large time frame to perform the attacks. Easier still would be to perform this attack remotely, via exploiting the Pass-Thru device's connected computer. However, when taking into account that servicing a vehicle is only a small fraction of a vehicle's lifespan, the overall window of opportunity is very small. Whilst some specialist knowledge is required to perform the attack, only knowledge of the software vulnerability within the Pass-Thru device is needed to modify the injected code. Some component knowledge is also needed to edit the malicious update code into the targeted system. Only a computer is required to perform this attack, without any peripheral equipment. The attack feasibility level of this attack is summarised in Table 11.

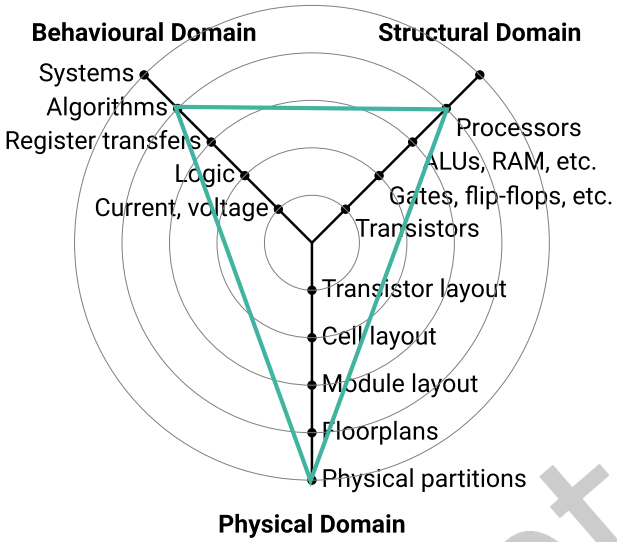

Fig. 4. Attack Hardware Classification: Update tampering. The attack targets the processors of the ECU, injecting code that is verified at the source and accepted by them. Algorithms are targetted via circumvention of regular update vetting and secure boot procedures.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Major	Major	Severe	Severe	Severe

Table 12. The impact of this attack is major, as compromising the vehicle may lead to its loss, incurring significant financial damages. The attack compromising the ECU source code is also a major privacy violation, both for the user and the manufacturer. Safety and operation are also greatly impacted thanks to complete overhaul of a potentially vital component. This means that every factor here is severely compromised.

If this attack is performed successfully, major components of the vehicle would be compromised completely, executing fully arbitrary code. This greatly impairs the safety of the vehicle. Financial impact is difficult to measure, but this may lead to the loss of the vehicle if functionality is significantly impaired. Important aspects of the vehicle may also be completely disabled, such as engine monitoring systems, preventing their operation. This may also allow for tracking of the vehicle if the GPS unit is tampered with via the update, and the reading of data from the head unit, including any linked devices such as a mobile phone. The impact ratings of this attack are summarised in Table 12.

4.3.3 Risk Mitigation. In this attack, a "pass-thru" device is connected via the OBD-II port. The OBD-II frames eventually go via the CAN network to the intended target. Each OBD-II frame has a CAN ID that is sent as part of the frame. Anomaly detection systems can be programmed to identify malicious CAN IDs that are specific OBD-II frames. Anomaly detection systems compare the real-time activity of a system against a recorded profile or rule. When a deviation from the rule is observed an alarm is raised.

As Lokman et al. [29] proposed anomaly-based intrusion detection system (IDS) done at the central gateway can help to identify spoof messages in the CAN bus. A compromised central gateway and a CAN network are a

greater threat compared to a compromised ECU. A safeguarded CAN network will safeguard the ECU. Therefore, in risk mitigation the asset that is focused on is the CAN network. Anomaly-based IDS is the most popular approach in automotive IDS. These malicious frames can be identified and blocked through an independent monitoring system such as Jadidbonab et al. [19] proposed Siemens Embedded Analytics IP. The additional information provided by the CAN Bus is used to identify the malicious frames. The analytical CPU will monitor all the traffic that is captured via the bus monitor and analyse any write access to memory registers that are forbidden.

As the analytical CPU is independent of the rest of the vehicle network it is not part of the compromised system thus enabling the detection and prevention. The register access writes which are required for the update tampering attack to succeed will be prevented by the analytical IP thereby mitigating against such an attack. A vehicle has a long lifespan of over 10 years and needs to be protected against threats unknown at the time of manufacturing. Anomaly based detection is capable of identifying novel attacks as an attack's characteristics are usually unknown at the time of manufacturing.

Anomaly-based IDS coupled with the hardware monitoring SoC will help to reduce the risk of an attack via a malicious Powertrain.

4.4 Spoofing Bus-Off attack against an ECU over CAN

4.4.1 Attack Description and Asset Identification. If an ECU cannot be easily compromised directly, either by tampering with updates or interacting with it physically, it can be compromised indirectly. Iehera et al. [16] use an indirect method to propose a masquerade attack against another ECU. A malicious device is introduced into the CAN network and spoofs legitimate messages in order to disconnect another legitimate ECU and prevent it from sending further messages on the bus via inducing error states. Therefore, the asset compromised in the attack is the CAN network.

An attacker introduces a custom ECU into the CAN network of the target vehicle. This ECU contains two components - a Bus-Off module to disable the target ECU, and a spoofing module to provide false messages from the target ECU on the CAN bus. Upon starting the vehicle, and just before the target ECU begins transmissions, a meaningless message is sent over CAN by the bus-off module. This message consists of one frame 255 bits long, to disallow the target ECU from sending messages whilst the frame is transmitting. The message has a higher identifier priority and is reporting an error state. When the receiving ECU reads this message, the Transmission Error Control counter for the target ECU is increased by 8, and then by a further 8 for every 14-bit period past the receiving time. When the Transmission Error Counter (TEC) hits 255, the ECU is transferred to the bus-off state - receiving and sending CAN messages is disabled for it. This attacking message is 255 bits long as this is the number of bits needed to achieve the bus-off state. Whilst the attacking message is being sent, bit-stuffing is performed in CAN to allow for spoofing messages to be sent, impersonating the target ECU. This allows for the complete replacement of the target ECU, and thereby control of its functionality. The attack vectors of this attack are summarised in Figure 5.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
1	6	3	4	7	21	Low

Table 13. This attack has a low feasibility of happening, as the equipment needed is quite bespoke, and the ability to place it into the vehicle is low and often requires another crime to put in, lowering the opportunity window. However, time to create a specialist fix for the exploit is fairly low.

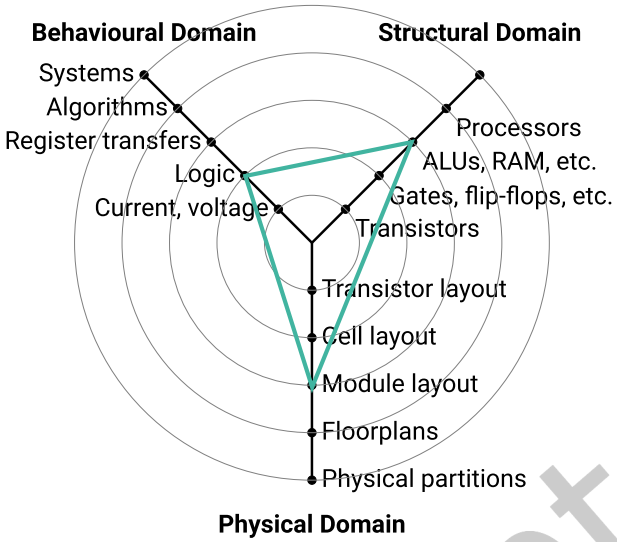

Fig. 5. Attack Hardware Classification: Spoofing Bus-Off attack. The attack targets the processor structure of the ECU, injecting a flood of error states that disallow legitimate messages to be properly sent. The logic of the processors is also hit.

4.4.2 Attack Feasibility and Impact Assessment. This attack would begin by adding the new ECU into the vehicle, which depending on the ECU compromised, may need a large window of opportunity to compromise. From there, the attack must happen while the vehicle is in motion, possibly triggered under certain conditions on the CAN bus. This means that once planted, the attack window is quite large. A custom ECU is needed, which may be difficult to acquire, and the subject of the bus-off attack must be studied extensively to know all messages expected from it. Specialists must also be needed to engineer the custom ECU and potentially insert it into the CAN bus. The attack feasibility level of this attack is summarised in Table 13.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Major	Major	Major	Moderate	Major

Table 14. This attack's impact is major as if the ECU is introduced into the vehicle, this foothold may allow the attacker to chain more exploits after the target ECU is disabled. This may lead to the loss of the vehicle, extraction of data impacting privacy or disabling of critical functionality causing safety impacts.

Preventing the function of a major ECU and falsely reporting values from it has a major impact on safety, potentially leading to the loss of major functions such as ABS. This may lead to an accident or collision. Privacy impact may come from leaking GPS information or similar but is otherwise negligible. Finally, this may cause the loss of the car if important features are impaired at critical moments, causing significant financial damage. The impact ratings of this attack is summarised in Table 14.

4.4.3 Risk Mitigation. This attack is carried out by using Bus-off module to disable the targetted ECU and introducing a malicious ECU into the network. In risk mitigation the asset that is focused on is the CAN network as it is the easiest point to detect the attack. As Cho et al. [11] identified the following two features makes identification of a bus-off attack easy:

- **F1**: at least 2 consecutive errors occurring during the tx frames with an active error flag.
- **F2**: at the time when the victim's Transmit Error Counter(TEC) increases a message with the same node ID is successfully transmitted by another ECU on the bus.

F1 is an indicator that the bus-off attack is happening and **F2** provides the required evidence. Once **F1** is seen on the monitoring system with a unique ID, the system can continue to monitor the CAN bus to check if any successful transmissions are seen with the same ID. This scenario can only occur if at least two ECUs are transmitting the same message simultaneously as the ID is unique to the ECU. This behaviour is not allowed in the CAN which leads to the possibility of a bus-off attack. A combination of **F1** and **F2** events verify the presence of a bus-off attack. Cho et. al's proposed method was able to observe the error frames on the CAN bus and prevent the bus-attacks efficiently as evidenced during the research stage of the countermeasure [11]. Jadidbonab et al. [19] proposed Siemens Embedded Analytics IP which is an independent monitoring system that provides a non-intrusive monitoring in hardware that can detect such behaviour and prevent any further messages from the compromised ECU. Once detected resetting the compromised ECU after a predetermined N consecutive error frames will help to bring the ECU out of that state [11]. This combined mitigation fully mitigates against spoofed bus-off attacks.

4.5 Tyre Pressure Monitoring System (TPMS) Tampering

4.5.1 Attack Description and Asset Identification. In-vehicle wireless connections are often used to communicate between components that cannot have a physical connection, due to moving components or other design constraints. However, in-vehicle wireless networks are much easier to remotely intercept and easily spoofable. They also allow much easier access to the ECU itself via communications. Rouf et al. [39] exploit the hardware of the TPMS here via reverse engineering of underlying communications protocols. This allows for remote error triggering, and a foothold into the in-vehicle network. Therefore, the asset compromised in the attack is the TPMS.

The TPMS ECU uses a short-range wireless communicator mounted on the ECU in order to communicate with the rest of the vehicle. This system, however, has very little security, with no authentication protocols or proper input validation. An attacker who is within a range of 40m is able to eavesdrop on the packets being sent to and from the TPMS system if the transmission frequency is known. Moreover, an attacker can send falsified packages using a hardware device capable of transmitting at 5GHz, with an underlying tone to prevent transmission of regular packages, with an effective range of 38m. This allows for remote compromising of the ECU traffic and provides a vehicle for more destructive attacks mentioned elsewhere in this report. The attack vectors of this attack are summarised in Figure 6.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
19	6	3	4	4	36	Very Low

Table 15. TPMS Tampering is classified as a very low likelihood of tampering, as whilst the communications can be tampered with at range, the range of 40m relegates this exploit to whilst the vehicle is in motion, lowering the opportunity window. Specialist interception technology is also needed.

4.5.2 Attack Feasibility and Impact Assessment. This attack would happen whenever the vehicle was turned on, but as the communications are only able to be intercepted within 40m, it must be done while the vehicle is in motion, reducing the opportunity window significantly. Expertise is needed in order to properly spoof communications and to set up short-range communications. Thorough knowledge of the communication protocols

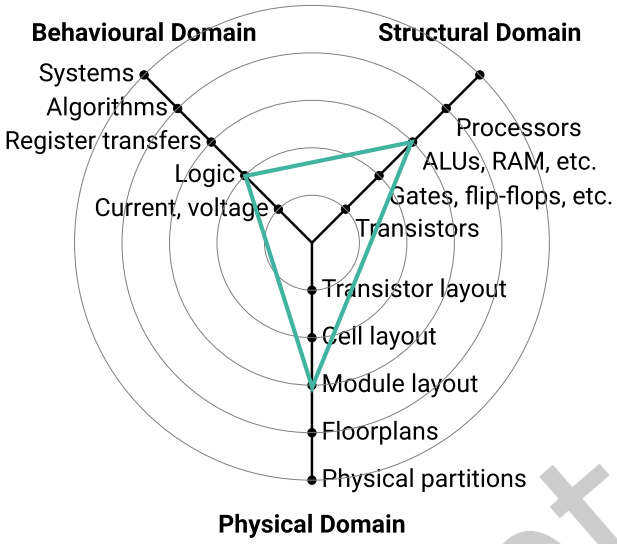

Fig. 6. Hardware attack classification: TPMS Tampering. This targets the communications modules of the TPMS chip. This also tampers with the logic of the communications, allowing legitimate messages to be spoofed at range. Finally, the module targets the processors, due to intercepting protocols and needing knowledge of the processor speed to do this.

is also necessary in order to spoof the TPMS messages and exploit the internal ECUs on reception. A specialist short-range communications rig is needed for the attack, as well as a way to keep up with the car while in motion if necessary. The attack feasibility level of this attack is summarised in Table 15.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Moderate	Moderate	Moderate	Negligible	Moderate

Table 16. This attack impact is moderate, as the foothold gained from compromising the TPMS communications is less impactful than other attacks mentioned here, allowing a limited window into the CAN network. Impacts on the wider vehicle are therefore limited because of this. PII is almost never compromised, and attacks resulting from this may be limited in scope to cause damage to operation and safety.

Depending on the ECU receiving the TPMS messages, exploitation may be possible in order to disable functions of the car, but may usually only affect tyre pressure readings. This may lead to an accident if a flat tyre affects the performance of the car. The financial impact is likely to only cost the value of the tyres on the vehicle, but operation may severely be limited by the loss of some control of the vehicle. Finally, the privacy impact is negligible. The impact ratings of this attack are summarised in Table 16.

4.5.3 Risk Mitigation. The TPMS attack is possible as the packets are transmitted via plaintext. The TPMS attack can be mitigated by obscuring the sensor ID. Therefore in order to mitigate the TPMS attack the mitigation will focus on the sensor as an asset. As Kilcoyne et al. [25] proposed by encrypting only the sensor ID in the TPMS protocol, this can be achieved. There is no need to encrypt the entire message. The sensor is the unique part of the TPMS message. By encrypting the rest of the message required diagnostics can still be carried on any tire

pressure issues. Using a linear feedback shift register (LFSR) is the most appropriate as no intense computation is introduced by its use of an XOR operation.

LFSR combines polynomials with feedback loops to create the encryption keys. A polynomial as shown below with 3 feedback loops can be used. TPMS messages are usually 32 bits. A 32-bit encryption polynomial LFSR with a 32-bit ID is vulnerable to brute force attacks. A polynomial with 2^{64} -1 phases provides adequate robustness to withstand brute force attacks by combing it with a 64-bit ID as follows:

$$2^{64} + 2^{63} + 2^{61} + 2^{60} + 1$$

The proposed set-up is:

(1) Encrypt sensor IDs using a LFSR phase.
(2) Using the matching phase and known stored ID values, the ECU conducts decryption.
(3) Diagnostic tools for maintenance are still able to read the messages. The sensor ID will appear as a random value.

Encryption is one mitigation recommended by Rouf et al. [39] who proposed the attack on TMPS. Encryption on its own is still exposed to replay and dictionary attacks. Including a sequence number in the payload would mitigate this weakness.

All suggested changes to the payload have the limitation that they require the TMPS sensors to have memory and the capabilities to conduct encryption which can be challenging. Therefore the mitigation will partially mitigate the attack.

4.6 Odometer Tampering

4.6.1 Attack Description and Asset Identification. Financial fraud is a major incentive to attack a vehicle, contrasting the major damage-focused methods. An attacker may use illegal means to compromise the odometer unit in order to reduce the mileage displayed. This increases the resale value of the vehicle via pretending it is newer, and thus may have less wear. Therefore, the asset compromised in the attack is the Odometer. Borkowski et al. [7] prove that this is not only accessible through a wide array of methods but also simple to circumvent security through commonly available tools.

In a modern digital odometer system, the value of the odometer is able to be edited in a number of ways, requiring more technical knowledge than in an older mechanical system. There are three main methods of editing one of these digital systems: Reprogramming the chip via physical access to the ECU, connecting a serial cable to the odometer unit itself, and connecting a servicing device to the OBD-II port. Reprogramming the chip is possible only with full disassembly of the odometer unit, using a common toolkit to enter the dashboard, gaining access to the unit, and connecting a recalibration device to the circuitry. Modification via attaching a serial cable is possible with disassembly of the dashboard. Little technical knowledge is required in order to modify the unit - recalibration devices for digital dashboards are available online. These are simple to use, and often are not manufacturer specific. The attack vectors of this attack are summarised in Figure 7.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
0	0	3	4	4	11	High

Table 17. This attack has a high feasibility of happening, due to the ease of execution for the attack via manufacturer tools, and the simplicity of the equipment used. In addition, the attack window is extremely large, and expertise necessary can be gained from numerous videos or websites freely available.

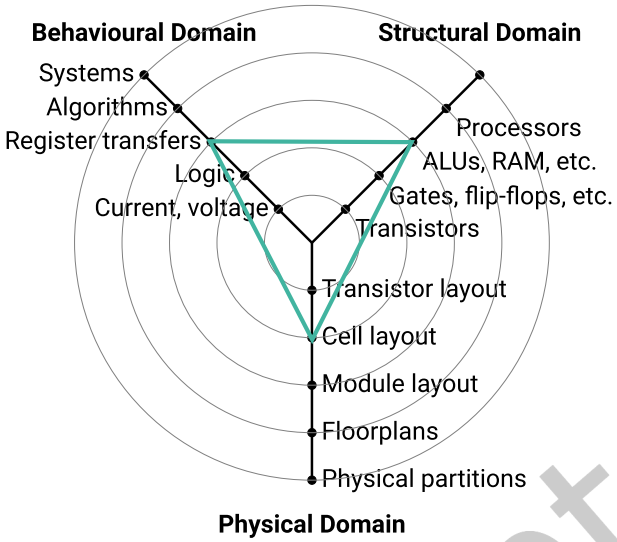

Fig. 7. Attack Hardware Classification: Odometer Tampering. The attack targets memory logic directly, editing values stored within. This directly affects the RAM via a physical connection. Physically, it targets specific modules (the memory) of the ECU.

4.6.2 Attack Feasibility and Impact Assessment. This attack would usually happen in a dealership, off the forecourt and before the car is sold, so the window of opportunity is very large. Very little specialist equipment is needed, and if it is, the equipment is freely available online. Very little expertise is needed - only knowledge to use a recalibration device. Component knowledge is only needed to find USB ports or manually rewind the odometer. The attack feasibility level of this attack is summarised in Table 17.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Moderate	Moderate	Moderate	Negligible	Moderate

Table 18. The attack impact is classified as Moderate, due to potential hazards arising from improperly serviced parts brought on by a falsified mileage count impacting operation of odometer and safety due to wear. Personal information and privacy is almost never impacted by this process, but financial losses may be incurred as a result of purchasing a fraudulently represented vehicle.

Safety impact may only affect the vehicle if the owner does not know how worn the components are via falsified lower mileages, potentially leading to more severe component failures. The resale value of the vehicle may be affected, and components may be replaced earlier than needed, due to falsified information in the sale. The operation may also suffer as a result of wear and tear. Privacy impact is negligible, as the attack often happens before the vehicle is sold and only affects the odometer. The impact ratings of this attack are summarised in Table 18.

4.6.3 Risk Mitigation. In a mileage tampering attack, the object is to disrupt the odometer value. By modifying the value in the CAN frame this attack is achieved. The odometer is the asset in focus for the risk mitigation.

Jadidbonab et al. [19] proposed Siemens Embedded Analytics IP is capable of identifying the modified CAN frames associated with mileage tampering. The proposed IP monitors the CAN Bus for CAN frames related to the vehicle's speed and odometer in real-time. The speed of the vehicle is extracted from the payload of the CAN frames related to speed. The timestamps of these CAN frames are also recorded.

Machine learning techniques are used to map between the miles travelled and the predicted odometer increments. Using this any deviations between the predicted and actual odometer increments are identified and raised. The CAN Bus is monitored for CAN frames containing the vehicle speed and odometer real-time values. Timestamps of the CAN frames are also captured. Vehicle speed is calculated from the payload of the CAN frames. The average speed of the vehicle over an observed time interval (Δt) is calculated. This value is fed into a linear transformation model that predicts the time delta (ΔT) of the unaltered mileage. The observed delta time (Δt) for the given speed is also calculated. Any deviations between ΔT and Δt outside of an error margin are flagged as an anomaly. The manipulated message is corrected by the analytics IP in the SoC.

The hardware monitoring SoC is able to detect and prevent when the mileage written on the odometer differs from the expected mileage due to vehicle speed. Thereby tampering of the odometer is prevented. Thus this control fully mitigates the odometer tampering attack.

4.7 Reverse engineering DST80-based Immobilizers

4.7.1 Attack Description and Asset Identification. Returning to the idea of vehicle theft, an important item to circumvent in the process of stealing the vehicle is the immobiliser, which prevents the unauthorized starting of the vehicle. Wouters et al. [51] pioneer a method of attack whereby the attacker uses power traces of the vehicle immobilizer combined with machine learning techniques in order to recover the randomly generated secure key from the DST80 encryption algorithm. This allows the starting of the vehicle without a legitimate key. Therefore, the asset compromised in the attack is the DST8-based Immobilizer.

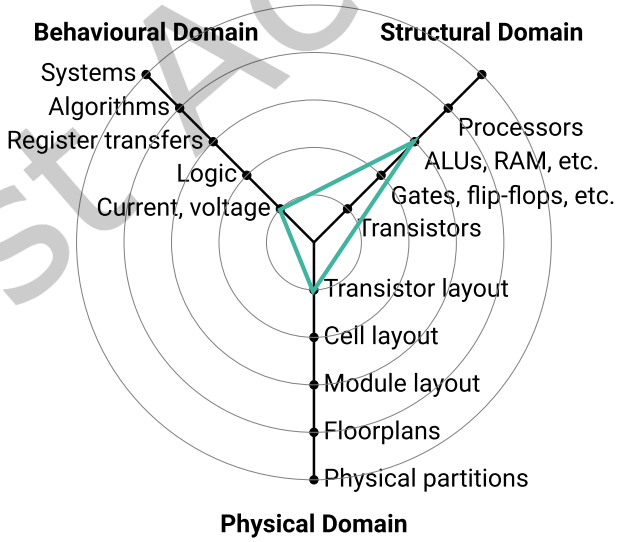

Fig. 8. Attack Hardware Classification: Disabling DST80-based immobilizers. The attack targets the voltage, utilizing power traces in order to reverse engineer the immobilizer. Transistor layout knowledge is also needed to measure specific power traces. Finally, the ALUs are exploited as these are the memory locations loading the key.

The key loading procedure of the immobilizer IC is standard whenever loading certain bits of the cypher. Each bit of the key is loaded sequentially before the DST80 cypher starts - therefore by analyzing the power trace of each load operation, an attacker is able to recover an unencrypted initial key with no prior knowledge of the encryption process. As the key is loaded in two parts of 40 bits each for a DST80-based system, this attack concentrates on retrieving one-half of the key.

With a pre-trained Multi-Layer Perceptron model using power traces from the target transponder chip, it becomes trivial to retrieve the key from the same transponder, essentially removing the need for a key to disable the immobilizer. With a sufficiently trained MLP model, it is possible to attack unknown systems that have no prior power traces recorded. This method, however, requires prior training of the model on the power traces of similar devices.

This retrieval of the key reduces the attack surface for the immobilizer from 2^{80} operations needed to 2^{40}, vastly reducing the time complexity of a successful attack of an unknown transponder. In addition, the number of power traces needed for a successful attack is reduced to 10. This number is realistically achievable in under 10 seconds of measurement. The attack vectors of this attack are summarised in Figure 8.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
1	6	3	4	4	18	Medium

Table 19. This attack has a medium feasibility of occurring, as the malicious hardware makes the attack easy in execution. However, the difficulties of getting physically close to the key, as well as the steep technological cost of the initial hardware, lower the feasibility.

4.7.2 Attack Feasibility and Impact Assessment. This attack would usually happen when the car is unlocked, allowing the key to be deciphered and the car to be retrieved later. As the model is able to be retrained, this increases the attack window significantly as no discovery is needed on traces for the immobilizer. A great deal of knowledge is needed on the general inner workings of the DST-80 cypher, and the specific power traces of that component in order to analyse how the key is calculated. The attack feasibility level of this attack is summarised in Table 19.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Moderate	Major	Major	Negligible	Moderate

Table 20. This attack has a moderate impact, as the occupant of the vehicle may be endangered or lose access to important contents if the vehicle is stolen. Vehicles often represent a steep financial investment as well, so the financial impact is major if it is stolen. Operational integrity of the immobilizer system is potentially removed or disabled, and safety of the vehicle's occupants is impacted thanks to this.

The attack would have a moderate impact on safety, leading potentially to the loss of the vehicle in a remote location. However, no other function of the car is affected besides access to the car itself. Again, this would lead to financial damages equivalent to the loss of the car itself. Loss of secure access to the car also constitutes major operational damage, not least if the car itself is stolen. Finally, privacy impact is moderate if car contents or information is lost. The impact ratings of this attack are summarised in Table 20.

4.7.3 Risk Mitigation. This attack uses the differential power analysis (DPA) technique to conduct a side-channel attack against the DST80. In risk mitigation the asset that is focused on is the DST80-based Immobilizer. As identified by Baddam et al. [4], varying both the voltage and frequency as a countermeasure for DPA attacks is not effective as frequency is easily detectable. Baddam et al. [4] proposed to keep the frequency constant while randomly varying the supply voltage.

A true random number generator (RNG) and a voltage controller are required for this mitigation. The mitigation can easily be applied to an ASIC or a general microcontroller without the need to modify the algorithm or design flow. For the mitigation to be effective the change in power consumed ($\delta_{voltage}$) due to the change in voltage (V_{dd}) needs to be close to the change in power consumed ($\delta_{switching}$) caused by input change.

Assume $[I_{n_1}, I_{n_2}, I_{n_3}, I_{n_4}, I_{n_5}]$ are input vectors and power consumed by each input at V_{dd_1} is $[P1_1, P1_2, P1_3, P1_4, P1_5]$ and $[P2_1, P2_2, P2_3, P2_4, P2_5]$ at V_{dd_2}. After I_{n_3} if the voltage was varied the resulting power consumption would be $[P1_1, P1_2, P1_3, P2_4, P2_5]$. As the difference between $P1_3$-$P2_4$ is dissimilar to $P1_3$-$P1_4$, this reduces the attack surface. Baddam et. al observed that the proposed method when applied with 10000 encryption rounds to AES reduced the correlation strength by 10 times. When applied to complete AES the secret key was indistinguishable thus preventing DPA attacks. For the mitigation to be effective a change in V_{dd} needs to appear as a change in input which is hard to achieve. Another limitation in this mitigation is, that if the attacker has direct access to the RNG and the voltage controller they can be disconnected exposing the system to a DPA attack. Given the limitations and sensitivity of the proposed mitigation, it is deemed to partially mitigate the attack.

4.8 Clock Skew

4.8.1 Attack Description and Asset Identification. Intrusion detection systems in a car, which act as a watchdog for internal systems and networks, are slowly gaining traction as a means to circumvent cyber threats to vehicles. Therefore, it is imperative that new attacks include a way to circumvent these systems on vehicles that include them. Therefore, the asset compromised in the attack is the IDS system. Sagong et al. [40] propose an intelligent attack that uses malicious ECUs within the CAN network, and mimics the clock frequency of legitimate devices in order to mask their own identity. This attack avoids Intrusion Detection Systems that would detect anomalies through bus traffic frequency. Through this method, the malicious ECU can better execute a bus-off attack, masking it's own identity under a better facade. Therefore, a foothold utilising a clock skew attack is a much stealthier way to execute an arbitrary damaging attack against a vehicle.

This attack is a more complex version of the bus-off attack mentioned previously, but modifications are made in order to fool smart Intrusion Detection Systems present in modern vehicles. These systems can approximate the normal behaviour of a system using metrics such as transmission intervals, timing and message contents. An attacker uses two previously compromised ECUs within the CAN network of the target vehicle as vessels for the bus-off attack, classified as a "strong" and "weak" attacker. The Strong attacker, on attack commencement, listens to the CAN transmissions by the target ECU. The interval between transmissions (T) is then calculated by the local clock of the strong attacker. After this gathering is complete, the bus-off attack begins. The strong attacker impersonates the target ECU, avoiding detection by sending spoofed messages every T seconds. Meanwhile, the weak attacker performs a one-frame bus-off attack, sending a message constructed of 255 bytes with a higher identifier precedence than the target ECU with the same ID. This trips the Transmission Error Control (refer to attack 4 for further details) and transitions the ECU to the bus-off state. This allows for complete spoofing of the target ECU. From this position, arbitrary CAN messages can be sent, disguised near-perfectly as the legitimate ECU targeted. Attacks launched from this foothold will differ depending on ECU targeted, but may involve suppressing engine temperature warnings, anti-lock brakes, and similar critical functions. The attack vectors of this attack are summarised in Figure 9.

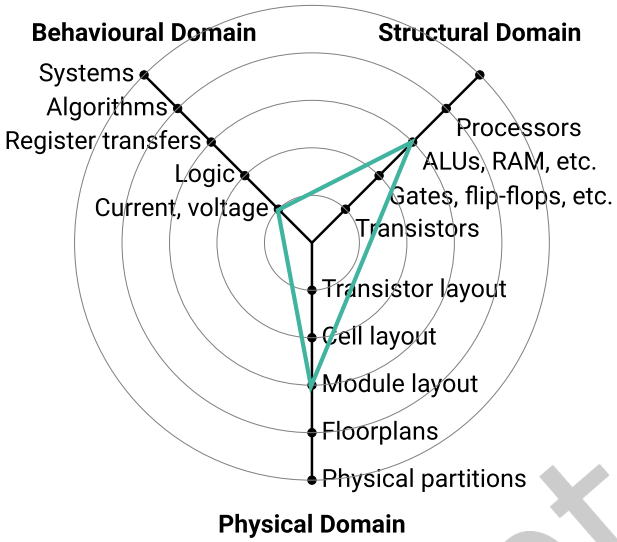

Fig. 9. Attack Hardware Classification: Clock Skew. The attack targets the logic of the ECU, looking to interrupt the transmission of messages to and from the processors by exploiting error logic. The processors are also targeted explicitly, allowing for interruption of normal operation. Finally, the module layout is targeted thanks to the exploitation of communications modules.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
4	6	3	4	4	21	Low

Table 21. This attack is classified as low feasibility, as the foothold needed into the CAN network in order to execute the attack coupled with the high degree of expertise needed makes the barrier to entry lower. Component knowledge is necessary but not to the same degree as previous bus-off attacks, thanks to the information-gathering phase automation items such as message delay. Equipment necessary is mostly limited to the malicious ECU.

4.8.2 Attack Feasibility and Impact Assessment. This attack would happen whilst the attack was in motion, using an ECU previously compromised. This complicates the attack window, as while this could happen anytime the vehicle is in operation, the window may enlarge or shrink depending on the method used to compromise the preliminary ECUs. A greater deal of component knowledge is needed than the previous bus-off attack, taking into account message timings and clock speeds of the target ECU. Experts are also necessary to first compromise an existing ECU, then properly tailor spoof messages to have an adverse effect on the vehicle. Finally, equipment needed is variable depending on the initial compromisation but is skewed towards specialist equipment due to the need to compromise 2 ECUs at least. The attack feasibility level of this attack is summarised in Table 21.

If this attack is successful, safety is severely compromised, especially through the attack's capability to avoid existing Intrusion Detection Systems. The financial impact may be caused by the loss of the vehicle, through compromising important ECUs such as engine management systems. Operation is severely compromised as key features of the vehicle may be completely disabled. However, the privacy impact here is negligible as it is unlikely data would be exfiltrated. The impact ratings of this attack are summarised in Table 22.

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Major	Major	Major	Negligible	Major

Table 22. This attack is classified with a major impact, as the foothold gained from both disabling a key ECU and evading existing detection mechanisms makes potential consequences for safety-critical systems extreme. However, the attack offers no way to impact the privacy of the road user, instead focusing on safety and operational impacts. Financial consequences here result from loss of vehicle, or loss of key components.

4.8.3 Risk Mitigation. In this attack proposed by Sagong et al. [40] a transmitter is used to manipulate the clock skew estimated by an IDS. The attacker therefore transmits every T = T+ΔT, where T is seconds and ΔT is the clock skew. Time-based IDS fails to mitigate this attack. In risk mitigation the asset that is focused on is the ECU via various mitigations.

Secure boot and monitoring of code executed from ROM memory are methods to mitigate this. Siddiqui's [43] proposed Core root of trust contains the executable code in the RAM which can be modified at boot time. The trusted platform module (TPM) contains a golden image to prevent attacks during the boot process. With the use of a Hash-based Message Authentication Code (HMAC) which is held in the Platform Configuration Registers (PCRs) of the TPM, a verification is conducted. Public and private keys are generated and the private key is sealed inside the TPM. During the boot process, a hash value is generated and is compared against the golden value. If it matches the values are changed. If not, countermeasures need to be taken. Jadidbonab et al. [19] proposed Siemens Embedded Analytics IP provides capabilities to detect and prevent any memory access to code within a protected region. The analytical CPU will compare memory address requests captured via the bus monitor and any requests for protected memory outside of the boot process will be blocked.

Through a combination of secure boot and hardware-based memory access monitoring and prevention, any malicious updates will be fully mitigated. The authors did not identify a specific mitigation technique to identify cloaked messages. However, the identification of malicious transactions mitigates the overall impact of the cloaked messages.

4.9 Reverse Engineering IOC

4.9.1 Attack Description and Asset Identification. Whilst the next attack is targeted against a specific vehicle, the extent of the attack combined with the devastating results makes it a worthy inclusion in this paper. Reverse engineering of the hardware within a Jeep Grand Cherokee allows for the multimedia chip within the head unit of the vehicle to be exploited. Therefore, the asset compromised in the attack is the multimedia chip. Miller and Valasek [31] prove that all systems within the car can be compromised via this foothold. From here, information within the head unit can be accessed, or further exploits can be employed to control functions such as steering, indicators or brakes.

Reverse engineering of the communication protocols of the OMAP chip within the head unit of the Jeep Grand Cherokee allows for a remote SSH shell to be set up via the D-Bus service on the chip. This allows for remote code execution on that chip. From here, it is possible to control the functionality of the head unit, to retrieve GPS or other private information. Reverse engineering of the memory structure of the v850 micro-controller chip allows for flashing of new firmware via the remote session - completely bringing the chip under the attacker's control. This reverse engineering is made possible by manufacturer diagnostic tools, freely available on the internet. By using these to retrieve ECU code, and identifying where this code is stored on-chip by use of in-system test functions provided by these tools, the attacker is able to target updates to manufacturer specifications. The use of these targeting techniques also allows the attacker to bypass code obfuscation via output monitoring

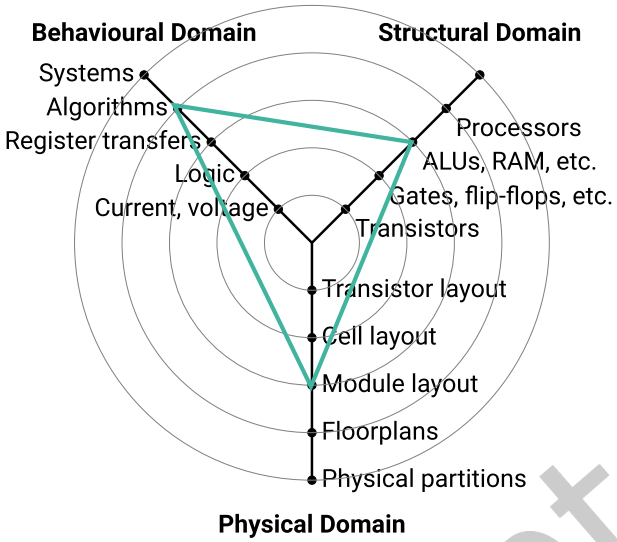

Fig. 10. Attack Hardware Classification: Reverse Engineering IOC. This attack targets algorithms vetting updates to the ECU, making the kill chain possible to execute. The ROM content is explicitly targeted in order to replace legitimate operating code with malicious equivalents, and the modules of the chip are separately targeted in order to hit specific memory locations.

techniques - put simply, decrypting code originally meant to be encrypted. Once the v850 chip is compromised, arbitrary messages can be sent over CAN, triggering responses from any chip on the network. This v850 chip and the Parking Assist Module (PAM) were also disabled in order to obtain the proper checksum algorithm and parameters needed to legitimize a CAN message on the network, allowing for a large degree of flexibility in controlling the vehicle's functions. The attack vectors of this attack are summarised in Figure 10.

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
4	6	11	0	4	25	Low

Table 23. This attack has a low feasibility of occurring, due to the high degree of component knowledge for multiple components of a specific vehicle needed to execute the attack. Whilst the attack can be executed remotely, making a high window of opportunity, the expertise needed lowers the feasibility to a low degree.

4.9.2 Attack Feasibility and Impact Assessment. This attack can happen completely remotely, meaning that the complete exploit chain can be executed from anywhere whilst the vehicle is turned on. This leads to an exceptionally large window of opportunity. A great deal of expertise is needed to properly execute the kill chain, chaining together multiple technically challenging exploits. Extensive component knowledge is also needed, combining update knowledge, memory structure and CAN protocols of multiple ECUs. However, performing the attack only requires a standard laptop. The attack feasibility of this attack is summarised in Table 23.

This attack allows for complete compromisation of the vehicle electronics, leading to a potential complete loss of control. If systems such as acceleration, braking or steering are compromised, this attack has severe and far-reaching consequences for safety. Key aspects of the vehicle may also be disabled completely until a complete

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Severe	Major	Severe	Major	Severe

Table 24. This attack is classified as having a major impact, due to the complete compromisation of the majority of the vehicle's function. This has the potential to compromise every facet of the vehicle operation, compromising safety through the steering and braking systems, extracting location and personal information, and completely disabling the car compromising operation. Loss of vehicle or insurance claims resulting from this breach also constitues major financial impact.

reset is done, thanks to update tampering. Data may be exfiltrated from the head unit, including GPS and PII information. Finally, this may lead to the loss of the vehicle, incurring significant financial damage. The impact ratings of this attack are summarised in Table 24.

4.9.3 Risk Mitigation. The attack based on Jeep Cherokee focused on exploiting a D-Bus message daemon running on port 6667 that accepted unauthenticated commands. The head unit which is the infotainment hub of the vehicle was targeted. Sprint and Cherokee released patches for this vulnerability that require authentication of messages via Telnet and blocking of any TCP/IP packet [31]. The multimedia chip within the head unit is the asset in focus for risk mitigation.

For any unpatched/vulnerable software still existing using an independent monitoring system such as Jadidbonab et al. [19] proposed Siemens Embedded Analytics IP can provide a mitigation. The analytics CPU is capable of identifying and preventing any forbidden access requests to the AXI addresses during normal operation or during start-up. The identification is done by comparing access requests to a set of pre-determined forbidden AXI addresses. The bus sentry will then prevent these transactions. The analytics IP coupled with the CAN Bus is also capable of identifying malicious CAN messages sent such as activating park-assist assistant in a travelling vehicle. Hardware-based monitoring is faster to respond in comparison to software monitoring.

The use of the analytics IP will fully mitigate even patched versions of the attack where an attack via Wi-Fi or a femtocell is still possible but has a low risk.

4.10 FPGA Hardware Trojans

4.10.1 Attack Description and Asset Identification. FPGA (Field Programmable Gate Array) are configurable ICs that are gaining a vast amount of popularity for many applications, including automotive processing. Mal-Sarka et al. [30] propose a new type of Trojan that is integrated into the simulated hardware of an FPGA, making it much more difficult to detect. Therefore, the asset compromised in the attack is the FPGA. A Trojan is a type of cyberattack that acts as a legitimate program until certain conditions are met to activate it. Attackers may compromise the FPGA at its inception, before insertion into a vehicle, to control functionality at critical intervals.

FPGA units within a vehicle may be compromised before insertion, and their underlying structure may be changed to perform differing operations under certain conditions. This attack is called a Hardware Trojan. Trojans may be triggered (activated) by either digital (signal-based) or analogue (environmental) factors and may provide digital or analogue responses. Digital responses may include modifying signals sent through interconnects or blocking signals from using said interconnects, and analogue responses may include overheating the FPGA or increasing system noise. Due to the configurable nature of FPGAs, it is noticeably easier to insert these attacks as opposed to modifying regular silicon, thus making them attractive targets for attacks. They may also be used for theft, by leaking decryption keys or passwords for IP. The attack vectors of this attack are summarised in Figure 11.

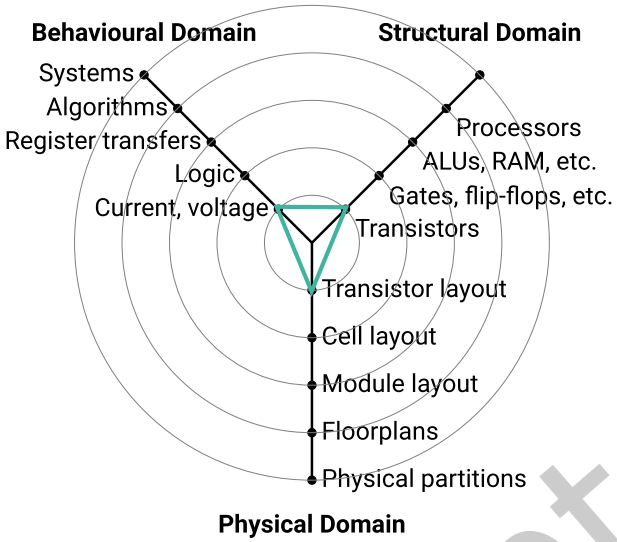

Fig. 11. Attack Hardware Classification: FPGA Trojans. Specific logic is targeted within the chip, including the gates and logical decisions to activate the trojan when environmental conditions are met. Transistors are targeted as the custom hardware implemented needs to be accurate down to the silicon, and the transistor placement is equally important

Elapsed Time	Specialist Expertise	Component Knowledge	Opportunity Window	Equipment	Total	Attack Feasibility Level
0	8	11	10	4	33	Very Low

Table 25. This attack has a very low feasibility of occurring, due to the incredibly small window of opportunity. The necessity of compromising the FPGA upon insertion makes ready access to the unit very difficult, and the hardware knowledge needing to be silicon-level increases the expertise needed incredibly. This is also a very recent find, so time to fix is negligible.

4.10.2 Attack Feasibility and Impact Assessment. This attack would happen within the supply chain, inserting the malicious FPGAs into the vehicle whilst in assembly. This reduces the opportunity window significantly, as access is needed into the assembly plant, potentially requiring insider or illegal access. A great deal of specialist expertise is necessary, encompassing the assembly processes, the structure of the simulated silicon, and how to make the additions not significantly impact the normal operation of the FPGA. Component knowledge of the original FPGA is also extensive, looking for spots to insert the modified hardware. Specialist software is also necessary, but a laptop should be all that is needed to modify the firmware. The attack feasibility level of this attack is summarised in Table 25.

If this is compromised, there would be a severe impact on the safety of the vehicle - FPGA trojans are incredibly difficult to detect, and may under certain conditions cause untold damage to the vehicle via disabling CAN traffic, causing adverse behaviour in systems, or reporting false information. This may lead to the loss of the vehicle and the financial damages associated with it. Furthermore, as FPGA systems are often bespoke to the vehicle, having one compromised may be an expensive loss in itself. Certain key operations of the vehicle may be completely disabled, potentially from a bus-off attack from the FPGA, or the cessation of function of the FPGA from the job

Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Level
Severe	Major	Severe	Moderate	Severe

Table 26. This attack is rated to have a major impact on the vehicle, as silicon layer compromisation has an incredibly large attack surface to compromise important facets of the vehicle. This extends to both important functions of the vehicle, and information such as location. Financial impact stems from potential loss of vehicle or contents due to failure. Safety and operation are both compromised from potential interruption of important vehicle function.

assigned to it. Privacy impact is less likely, but the trojan may exfiltrate data or compromise GPS traffic. The impact ratings of this attack are summarised in Table 27.

4.10.3 Risk Mitigation. In risk mitigation the asset that is focused on is the FPGA.A triple modular redundancy (TMR) is a popular mitigation technique for FPGA trojans. A TMR will consist of 3 copies of the original circuit and majority voter. Using a comparator circuit, the outputs of the two instances are compared and if a mismatch occurs then the third circuit and the arbiter circuit is enabled. Looking at an instance where 3 adders (O_i,R_i,S_i) are mapped to the Trojan-infected FPGA, their outputs will be compared via a comparator. Mal-Sarkar et al. [30] proposed Adapted Triple Modular Redundancy (AMTR), which implements variants of the 2 adders O_i and R_i required in the TMR instead of 3. By implementing variants both adders will functionally compute the same results although structurally they are different. Structurally different adders can be implemented by differentiating synthesis constraints such as the size and latency of instances of the same module.

To trigger both adders by an adversary will require implementing different logic blocks, storage elements, memory locations and interconnects. Due to this complexity, an attacker triggering the same Trojan via both adders is very rare. Given triggering the same Trojan via variants is rare this mitigation is effective in blocking hardware trojan attacks. Therefore an ATMR fully mitigates against FPGA hardware Trojan attacks.

4.11 Summary Table

A summary is presented below of the attack vectors listed above, their rating and overall impact assessment.

5 CONCLUSION

Whilst the attacks in this paper exploit hardware, there are a great deal of differing methods of attack. Some require physical access to the vehicle, whilst some are fully remote. However, some patterns emerge in the behaviour of these attacks. All listed attacks here target specific ECUs, with the intent of either manipulating or disabling their communications to other components in the vehicle or outside of it. The attacks also either aim to harm the vehicle, or gain data that otherwise should be private or protected, such as encryption keys. The most damaging attacks in the first category tend to be those that are the closest to the silicon structure of the vehicle, including targeting specific memory locations or ECU electronics.

Manipulation of ECU updates is also a significant trend in the most damaging attacks, with those that manage to control the update process gaining the greatest success in modifying other aspects of the vehicle. The most damaging attacks also tend to have the smallest window of opportunity, requiring privileged access to the vehicle. This, unfortunately, has one major exception as shown in Section 4.9.

This paper has paved the way for a systematic risk characterisation of hardware attacks. The use of Gajski-Kuhn Y-chart [22] to identify what domain of hardware is being manipulated helps towards a deeper understanding of the attack vector being deployed, and also the potential mitigation to be deployed. As such, behavioural, structural and physical domains of hardware are fundamentally diverse, yet highly relevant to characterise the nature of an attack, which may need effecting across different domains in a coordinated fashion.

Attack	Safety Impact	Financial Impact	Operational Impact	Privacy Impact	Overall Impact Level
EMFI	Severe	Major	Severe	Severe	Severe
Update Tampering	Major	Major	Severe	Severe	Severe
Reverse Engineering	Severe	Major	Severe	Major	Severe
FPGA Trojan	Severe	Major	Severe	Moderate	Severe
Bus-Off Attack	Major	Major	Major	Moderate	Major
Clock Skew	Major	Major	Major	Negligible	Major
Reverse Engineering key fob	Moderate	Major	Moderate	Major	Major
DST-80 Immobilizers	Moderate	Major	Major	Negligible	Moderate
TPMS Tampering	Moderate	Moderate	Moderate	Negligible	Moderate
Odometer Tampering	Moderate	Moderate	Moderate	Negligible	Moderate

Table 27. We summarise all ten attacks, where for each attack we present the estimated impact for safety, financial cost, operational disruption and loss of privacy. The overall impact level is based on the highest impact from any of the four categories above. The table is ordered by decreasing severity of overall impact.

The mitigations proposed in the paper focused on identifying a single attack point and safeguarding it to ensure the attack was not successful. Identifying mitigations that are non-intrusive and require minimal changes to existing infrastructure is challenging. Mitigations that suggested modifications to a data packet format have been disregarded as they require modifications to the existing hardware in vehicles. Smart factories share commonalities with the automotive industry as smart factories are heavily sensor-based and contain a lot of data digitally.

The majority of the attacks that target an ECU or a sensor can be detected and prevented by a hardware monitoring SoC. The enhanced CAN-HG^{TM} bus controllers provide the SoC with the required information to conduct the detection. A hardware monitoring system is non-intrusive and does not require changes to any existing components in the vehicle as it's an independent SoC. Encrypting the data sent and data validation are other key mitigations that will help increase the security of a vehicle.

In the future, we will follow several directions from our results. First, we aim to develop a supporting tool to systematise the characterisation process of analysing and evaluating risks caused by hardware attacks. Such a tool will facilitate the process for cybersecurity experts and engineering practitioners when facing zero-day and uncharacterised hardware attacks. The systematic characterisation aspect of the process will ensure the objectiveness of the characterisation of any hardware attacks. Second, we will construct an open-source library of threat models for hardware attacks. Each threat model can be formally represented by an attack-defense tree [28], which will be useful for further security analysis. For example, security analysts can use them to construct a system-wise threat model [20] for an entire system design. Finally, the characterisation of hardware attacks can provide faithful simulations integrated into existing digital twins. This enables one to simulate attack scenarios realistically for objective security analysis and impact evaluation.

ACKNOWLEDGMENTS

Contribution by Hoang Nga Nguyen and Siraj Ahmed Shaikh has been supported by AutoCHERI (https://autocheri.tech/), (InnovateUK project reference 10018347), which is funded under the Digital Security by Design (DSbD) initiative addressing secure by design principles.

REFERENCES

[1] Wim Aerts, Eli Biham, Dieter Moitie, Elke Mulder, Orr Dunkelman, Sebastiaan Indesteege, Nathan Keller, Bart Preneel, Guy Vandenbosch, and Ingrid Verbauwhede. 2012. A Practical Attack on KeeLoq. *J. Cryptology* 25 (01 2012), 136–157. https://doi.org/10.1007/s00145-010-9091-9

[2] Zeyad Al-Odat, Mazhar Ali, and Samee U. Khan. 2018. Mitigation and Improving SHA-1 Standard Using Collision Detection Approach. (2018), 6.

[3] Aiman Al-Sabaawi, Khamael Al-Dulaimi, Ernest Foo, and Mamoun Alazab. 2021. Addressing malware attacks on connected and autonomous vehicles: recent techniques and challenges. *Malware Analysis Using Artificial Intelligence and Deep Learning* (2021), 97–119.

[4] Karthik Baddam and Mark Zwolinski. 2007. Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure. (2007), 6.

[5] Sebastian Bittl, Arturo A. Gonzalez, Matthias Myrtus, Hanno Beckmann, Stefan Sailer, and Bernd Eissfeller. 2015. Emerging attacks on VANET security based on GPS Time Spoofing. In *2015 IEEE Conference on Communications and Network Security (CNS)*. 344–352. https://doi.org/10.1109/CNS.2015.7346845

[6] Vilas Boas, Chris C Dao, and Stefano Pietri. 2012. United States Patent USOO8253453B2.

[7] Przemyslaw Borkowski. 2019. Reducing odometer fraud in the EU second-hand passenger car market through technical solution. In *Integration as Solution for Advanced Smart Urban Transport Systems: 15th Scientific and Technical Conference "Transport Systems. Theory & Practice 2018", Selected Papers"*. Springer, 184–194.

[8] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. 2019. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418* (2019).

[9] Efstratios Chatzoglou, Georgios Kambourakis, and Vasileios Kouliaridis. 2021. A Multi-Tier Security Analysis of Official Car Management Apps for Android. *Future Internet* 13, 3 (2021). https://doi.org/10.3390/fi13030058

[10] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al. 2011. Comprehensive experimental analyses of automotive attack surfaces.. In *USENIX security symposium*, Vol. 4. San Francisco, 2021.

[11] Kyong-Tak Cho and Kang G. Shin. 2016. Error Handling of In-vehicle Networks Makes Them Vulnerable. (2016), 12.

[12] Ludvig Christensen and Daniel Dannberg. 2019. Ethical hacking of IoT devices: OBD-II dongles.

[13] Riccardo Coppola and Maurizio Morisio. 2016. Connected Car: Technologies, Issues, Future Trends. *ACM Comput. Surv.* 49, 3, Article 46 (oct 2016), 36 pages. https://doi.org/10.1145/2971482

[14] Zeinab El-Rewini, Karthikeyan Sadatsharan, Niroop Sugunaraj, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. 2020. Cybersecurity attacks in vehicular sensors. *IEEE Sensors Journal* 20, 22 (2020), 13752–13767.

[15] Sibylle Fröschle and Alexander Stühring. 2017. Analyzing the Capabilities of the CAN Attacker. In *Computer Security – ESORICS 2017*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer International Publishing, Cham, 464–482.

[16] Kazuki Iehira, Hiroyuki Inoue, and Kenji Ishida. 2018. Spoofing attack using bus-off attacks against a specific ECU of the CAN bus. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–4.

[17] ISO. 2021. *Document management—Portable document format—Part 1: PDF 1.7*. ISO/SAE 21434:2021. International Organization for Standardization, Geneva, Switzerland.

[18] Viacheslav Izosimov, Alexandros Asvestopoulos, Oscar Blomkvist, and Martin Törngren. 2016. Security-aware development of cyber-physical systems illustrated with automotive case study. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 818–821.

[19] Hesamaldin Jadidbonab, Hoang Nga Nguyen, Siraj Ahmed Shaikh, Marcin Hlond, Peter Robertson, and Gajinder Panesar. 2022. A Hardware-based SoC Monitoring In-life Solution for Automotive Industry. (2022), 6.

[20] Hesamaldin Jadidbonab, Hoang Nga Nguyen, Siraj Ahmed Shaikh, Marcin Hlond, Peter Robertson, and Gajinder Panesar. 2022. A Hardware-based SoC Monitoring In-life Solution for Automotive Industry. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom 2022 Workshops, Pisa, Italy, March 21-25, 2022*. IEEE, 637–642. https://doi.org/10.1109/PERCOMWORKSHOPS53856.2022.9767491

[21] Ahmer Khan Jadoon, Licheng Wang, Tong Li, and Muhammad Azam Zia. 2018. Lightweight Cryptographic Techniques for Automotive Cybersecurity. *Wireless Communications and Mobile Computing* 2018 (June 2018), 1–15. https://doi.org/10.1155/2018/1640167

[22] Hubert Kaeslin. 2015. *Top-down Digital VLSI Design: From Architectures to Gate-Level Circuits and FPGAS.* Morgan Kaufmann is an imprint of Elsevier, Waltham, MA.
[23] Trishank Karthik, Akan Brown, Sebastien Awwad, Damon McCoy, Russ Bielawski, Cameron Mott, Sam Lauzon, André Weimerskirch, and Justin Cappos. 2016. Uptane: Securing Software Updates for Automobiles. In *International Conference on Embedded Security in Car*. 1–11.
[24] Shah Khalid Khan, Nirajan Shiwakoti, Peter Stasinopoulos, and Yilun Chen. 2020. Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions. *Accident Analysis & Prevention* 148 (2020), 105837.
[25] Deirdre K. Kilcoyne, Shiri Bendelac, Joseph M. Ernst, and Alan J. Michaels. 2016. Tire Pressure Monitoring System Encryption to Improve Vehicular Security. (2016), 6.
[26] Kyounggon Kim, Jun Seok Kim, Seonghoon Jeong, Jo-Hee Park, and Huy Kang Kim. 2021. Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security* 103 (2021), 102150. https://doi.org/10.1016/j.cose.2020.102150
[27] Rony Komissarov and Avishai Wool. 2021. Spoofing Attacks Against Vehicular FMCW Radar. In *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security* (Virtual Event, Republic of Korea) *(ASHES '21)*. Association for Computing Machinery, New York, NY, USA, 91–97. https://doi.org/10.1145/3474376.3487283
[28] Barbara Kordy, Sjouke Mauw, Sasa Radomirovic, and Patrick Schweitzer. 2010. Foundations of Attack-Defense Trees. In *Formal Aspects of Security and Trust - 7th International Workshop, FAST 2010, Pisa, Italy, September 16-17, 2010. Revised Selected Papers (Lecture Notes in Computer Science, Vol. 6561)*, Pierpaolo Degano, Sandro Etalle, and Joshua D. Guttman (Eds.). Springer, 80–95. https://doi.org/10.1007/978-3-642-19751-2_6
[29] Siti-Farhana Lokman, Abu Talib Othman, and Muhammad-Husaini Abu-Bakr. 2019. Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. *EURASIP Journal on Wirless Communications and Networking* (2019), 17.
[30] Sanchita Mal-Sarkar, Robert Karam, Seetharam Narasimhan, Anandaroop Ghosh, Aswin Krishna, and Swarup Bhunia. 2016. Design and validation for FPGA trust under hardware trojan attacks. *IEEE Transactions on Multi-Scale Computing Systems* 2, 3 (2016), 186–198.
[31] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* 2015, S 91 (2015), 1–91.
[32] Athanasios Moschos, Kevin Valakuzhy, and Angelos D Keromytis. 2022. On the Feasibility of Remotely Triggered Automotive Hardware Trojans. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 1–6.
[33] Dennis K. Nilsson, Ulf E. Larson, and Erland Jonsson. 2008. Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles. In *Computer Safety, Reliability, and Security*, Michael D. Harrison and Mark-Alexander Sujan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 207–220.
[34] Colin O'Flynn. 2020. BAM BAM!! On Reliability of EMFI for in-situ Automotive ECU Attacks. *Cryptology ePrint Archive* (2020).
[35] Irdin Pekaric, Clemens Sauerwein, Stefan Haselwanter, and Michael Felderer. 2021. A taxonomy of attack mechanisms in the automotive domain. *Computer Standards & Interfaces* 78 (10 2021), 103539. https://doi.org/10.1016/j.csi.2021.103539
[36] B. Phan, F. Mannan, and F. Heide. 2021. Adversarial Imaging Pipelines. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 16046–16056. https://doi.org/10.1109/CVPR46437.2021.01579
[37] Vishnu Renganathan, Ekim Yurtsever, Qadeer Ahmed, and Aylin Yener. 2022. Valet Attack on Privacy: A Cybersecurity Threat in Automotive Bluetooth Infotainment Systems. *Cybersecurity* 5, 1 (Oct. 2022), 30. https://doi.org/10.1186/s42400-022-00132-x
[38] David Rogers. 2021. *Effectively addressing the challenges of securing connected and autonomous vehicles.* Technical Report. Copper Horse.
[39] Ishtiaq Rouf, Robert D Miller, Hossen A Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. 2010. Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study.. In *USENIX Security Symposium*, Vol. 10.
[40] Sang Uk Sagong, Xuhang Ying, Andrew Clark, Linda Bushnell, and Radha Poovendran. 2018. Cloaking the clock: Emulating clock skew in controller area networks. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 32–42.
[41] Xhani Marvin Saß and Richard Mitev. 2023. How to Multi-Glitch the Glitching-Protections on ARM TrustZone-M. *arXiv arXiv:2302.06932* (2023), 18.
[42] Siddharth Shukla. 2016. *Embedded Security for Vehicles : ECU Hacking.* Master's thesis. Uppsala University, Department of Information Technology.
[43] Ali Shuja Siddiqui. 2020. *DESIGN OF SECURE BOOT PROCESS FOR RECONFIGURABLE ARCHITECTURES.* Ph. D. Dissertation. The University of North Carolina.
[44] Fahad Siddiqui, Rafiullah Khan, Sena Yengec Tasdemir, Henry Hui, Balmukund Sonigara, Sakir Sezer, and Kieran McLaughlin. 2023. Cybersecurity Engineering: Bridging the Security Gaps in Advanced Automotive Systems and ISO/SAE 21434. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. 1–6. https://doi.org/10.1109/VTC2023-Spring57618.2023.10200490
[45] Christoph Sommer and Falko Dressler. 2015. *Vehicular Networking.* Cambridge University Press, Cambridge, United Kingdom.
[46] Bas GB Stottelaar. 2015. *Practical cyber-attacks on autonomous vehicles.* Master's thesis. University of Twente.
[47] Andrew Tomlinson, Simon Parkin, and Siraj Ahmed Shaikh. 2022. Drivers and barriers for secure hardware adoption across ecosystem stakeholders. *J. Cybersecur.* 8, 1 (2022). https://doi.org/10.1093/cybsec/tyac009

[48] UNECE. 2021. United Nations Regulation No. 155 - Cyber Security and Cyber Security Management System. https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security. Accessed: 2023-08-30.
[49] Ingrid Verbauwhede. 2021. Hardware Security Knowledge Area Version 1.0.1.
[50] Lennert Wouters, Benedikt Gierlichs, and Bart Preneel. 2021. My other car is your car: compromising the Tesla Model X keyless entry system. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021, 4 (2021), 149–172.
[51] Lennert Wouters, Jan Van den Herrewegen, Flavio D Garcia, David Oswald, Benedikt Gierlichs, and Bart Preneel. 2020. Dismantling DST80-based immobiliser systems. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 2 (2020), 99–127.
[52] Chen Yan. 2016. Can You Trust Autonomous Vehicles : Contactless Attacks against Sensors of Self-driving Vehicle. https://api.semanticscholar.org/CorpusID:27264520

Received 31 August 2023; revised 29 January 2024; revised 19 March 2024; accepted 8 April 2024