EDA for Secure and Dependable Cybercars: Challenges and Opportunities

Farinaz Koushanfar
Electrical& Computer Engineering
Rice University
Houston, TX
farinaz@rice.edu

Ahmad-Reza Sadeghi[†], Hervé Seudie[†,‡]
[†]Fraunhofer SIT, Germany
[†]Intel-TU Darmstadt Security Institute, Germany
[‡]Robert Bosch GmbH, Germany
{ahmad.sadeghi,herve.seudie}@trust.cased.de

ABSTRACT

Modern vehicles integrate a multitude of embedded hard realtime control functionalities, and a host of advanced information and entertainment (infotainment) features. The true paradigm shift for future vehicles (cybercars) is not only a result of this increasing plurality of subsystems and functions, but is also driven by the unprecedented levels of intra- and inter-car connections and communications as well as networking with external entities.

Several new cybercar security and safety challenges simultaneously arise. On one hand, many challenges arise due to increasing system complexity as well as new functionalities that should jointly work on the existing legacy protocols and technologies; such systems are likely unable to warrant a fully secure and dependable system without afterthoughts. On the other hand, challenges arise due to the escalating number of interconnections among the real-time control functions, infotainment components, and the accessible surrounding external devices, vehicles, networks, and cloud services. The arrival of cybercars calls for novel abstractions, models, protocols, design methodologies, testing and evaluation tools to automate the integration and analysis of the safety and security requirements.

Categories and Subject Descriptors

C.3 [**Real-time and embedded systems**]; D.4.6 [**Software**]: OPERATING SYSTEMS—*Security and Protection*

General Terms

Security, Reliability

Keywords

Perspective Article, Automotive Security, CPS Security

1. INTRODUCTION

Modern vehicles include several Electronic Control Units (ECUs) that form an in-vehicle distributed networked embedded system. The ECU networks not only command the hard realtime control of automobile mechanical parts and support infotainment functions [14], but also they provide a gateway between modern cars and their surroundings (e.g., traffic lights), devices (e.g., smartphones), vehicles, and accessible networks. The terms car-to-car and car-to-X (infrastructure or device) are used to refer to cybercars' communication scenarios. The emergence of Intelligent Transport System (ITS) is expected to further reduce the number of road accidents and improve the road traffic conditions. Furthermore, connecting cars to the cloud or smartphones offers a new set of applications and business models. In this context, Infotainment and safety related subsystems may need to interact to provide various information to the driver.

Figure 1 shows the view of ITS depicted by the standardization body ETSI (European Telecommunication Standards Institute). While the introduction of new communication technologies offers an unprecedented number of new opportunities, it increases the complexity of the car system and demands new analysis of security and privacy requirements. A few preliminary attacks that could seriously impact the car's safety and reliability have already been demonstrated in simulations or experiments including [26, 37, 27, 28, 43, 31, 12]. For example, a study by Barisani et al. shows that 2 malicious cars out of 400 vehicles affected 20% of the traffic [9], where this estimation did not even consider system failures due to design errors or poor implementation and testing schemes. A comprehensive summary of related work which includes the description of the demonstrated attacks, is provided in Appendix A.

The cyber-physical attributes of modern automotive systems directly link security vulnerabilities to the cybercar's physical safety and reliability features. Therefore, the scope of the potential vulnerabilities is much more vast than what has been demonstrated, and is far beyond attacking an individual car [42, 29]. Thinking of vehicle safety without considering security, as it was done in the past, is no longer a viable option. In this respect, security vulnerabilities of cars are markedly different from typical security issues in conventional computer and network systems.

A few rather recent projects and initiatives have started investigating the safety and security of modern vehicular electronics, including [51, 52, 20]. The work thus far has mostly centered on identifying protection primitives to be included in the emerging standards, characterizing the attack models and security threats, as well as devising technical and specific protection guidelines to further secure cybercars. Independent of the cybercar security initiatives, the

DAC 2012, June 3-7, 2012, San Francisco, California, USA.

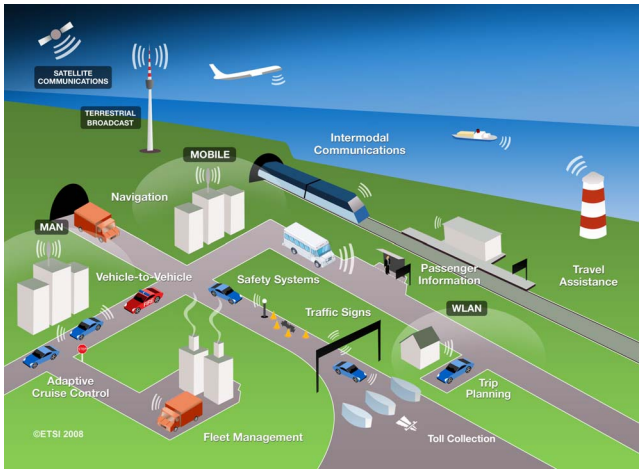

Figure 1: Intelligent Transport Systems. Source: www.etsi.org

EDA and embedded systems communities have been working for years on problems pertaining to modeling, analysis, simulation, and automation of complex vehicular networked embedded systems. Such methods are required not only to ensure design time predictability and composability, but also for architecture selection and design-space exploration [48].

The complexity of the modern cybercar's networked embedded systems, its various interfaces to external entities, together with the scope of emerging attacks, supersedes present knowledge and capabilities pertaining to both lines of efforts in vehicular security and in embedded system communities. The evolving nature of the system complexity and attack possibilities suggests that a continuous flow of research and development is needed before cybercar systems can be efficiently designed, and safely/securely operated.

The situation with cybercars today is similar to the evolution of personal computing and networked communication in the past several decades: One can make an analogy between connecting individual cars to external objects/networks and linking personal computers to the Internet. Since the Internet was not originally designed with an explicit set of security objectives, connected computers still suffer from a range of attacks that could be largely avoided by correct-by-construction methodologies. Due to safety criticality and the vital role of vehicular and transportation systems in personal, business, government, and economic affairs, leaving security as an afterthought is disadvantageous. However, since legacy protocols and hardware take a long time to change, for present and pending cybercar generations, security afterthoughts maybe the only practicable choice.

This perspective article calls for development of novel holistic but systematic EDA methodologies and tools that simultaneously ensure a robust and secure cybercar design flow. The important architecture-evaluation, design, and evaluation process phases need to be automatically augmented with security primitives and flows as defined by rising standards and protection measures. The challenges for realizing such a holistic and systematic automated cybercar security approach are abundant, but so are the research and development opportunities.

2. PRELIMINARIES

Before we delve into more detailed discussions of cybercars security and the pertinent challenges and opportunities, we briefly outline a typical cybercar system architecture and some key evolving standards.

Cybercar system architecture. The term cybercar refers to a generation of vehicles that are fully connected to their surrounding objects, environments, and networks. A cybercar is part of an ecosystem where it could either play the role of a content or a service provider in addition to expressing content and determining applications. Thus, a cybercar can be described as a sophisticated mobile device. As it is connected to external entities, the cybercar has the possibility to rely on outside computing resources, externally available data, and services. For instance, a cybercar without the proper sensors could still run driving assistance applications with the help of a smartphone and a cloud connection. A cybercar may also provide services to connected entities. Cybercars are composed of following domains: (i) the in-car system with the network and the car components, (ii) communication interfaces to external communication partners, (iii) communications partners represent by external devices, cars, infrastructure or cloud. Figure 2 represents an example of a cybercar system architecture.

Emerging automotive standards. With some exceptions [4, 2], automotive standards are only regionally accepted, which might be due to the role of the legislation and the influence of regional automotive industry. However, the AUTOSTAR experience has shown that the global participation of automotive manufacturers and suppliers coming from different countries in the specification process, is a key to the global acceptance of standards [4].

With the emergence of cybercars and the requirement of global acceptance, the automotive industry has created different consortiums to support the specification of standards for ITS. Examples are the GENIVI alliance, the Car Connectivity Consortium and the Car2Car Consortium [7, 13, 55], where the first two are driving the specification of an in-vehicle infotainment platform with different connectivity technologies. The Car2Car Consortium coordinates different investigation results on communication and system architectures for car-to-car and car-to-infrastructure communication. The resulting specifications are provided to standardization bodies. The Sevecom and the EVITA projects are examples of unclassified funded projects [20, 51] that have significantly influenced standardization bodies like the ETSI and the Communication Access for Land Mobiles (CALM) [6, 5].

Typical goals of the automotive industry are guaranteeing interoperability, reliability, dependability and quality. Security has not been a major concern. Hence, standards for the car components and networks have been specified with very limited security requirements. The integration of IT in cars and recent attacks on cars has led to a change of this view [12]. As a result, the security has been integrated both in cybercar related standardization activities and in established specifications such as AUTOSAR, with the introduction of cryptographic interfaces [4]. There is still a need to analyze the overall impact of new technologies emerging on the automotive process, development, and tool chain, and the role of security.

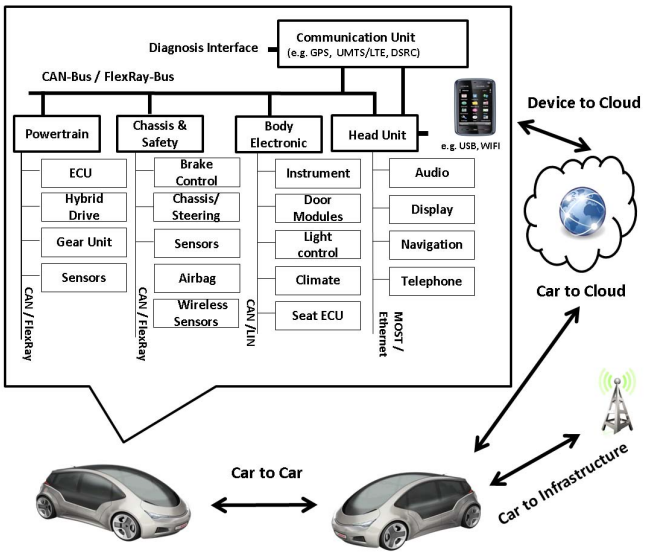

Figure 2: Cybercar System Architecture.

3. CYBERCAR SECURITY CHALLENGES

3.1 Threats

The automobile industry has traditionally focused on providing safety and reliability, under the assumption that a vehicle is an isolated system which is not accessible to adversaries [35], leading to the design of insecure car components and bus systems. Several successful attack scenarios have invalidated traditional models for cybercar security [26, 37, 43, 11, 19, 31, 12, 56]. The vulnerabilities are exasperated by poor implementation of protocols and firmware [30, 39]. The growing connectivity of modern cybercars escalates the range of potential exploits [42, 29, 18]. The existing weaknesses in the automotive domain can be classified as follows:

- Threats caused by physical access, e.g., by connecting to the in-car network through the on-board diagnostic interface, or the in-vehicle network cables.
- Vulnerabilities due to access via infotainment, which introduce the possibility to manipulate cars using multimedia interfaces, e.g., by disabling safety functions.
- Exposures due to the remote access, which describe the possibility to manipulate cars using wireless interfaces.

Exploiting such vulnerabilities may even enable the complete remote control of a car by an attacker, who might possess the expertise to gain access to the in-vehicle network using the on-board diagnostic interfaces or the multimedia interfaces (e.g., USB connection, Bluetooth connection or media players). Such an adversary could remotely disable brakes or manipulate sensor values using typical attacks such as eavesdropping, dropping, modification, spoofing, injection, or message replay on the bus system [43, 12].

3.2 Challenges

The security and dependability challenges of cybercars are closely related to the general engineering challenges of cars and the current state of car networks. These challenges include,

- Automotive components have resource constraints, e.g., limited memory, processing, or number of sensors.
- Automotive networks are insecure and have limited throughput with strict latency requirements.
- Automotive components must be cost effective. Security must be integrated without a high cost overhead.
- The lifecycle of the automotive industry is up to 20 years. Solutions should be capable to hold for a long time.
- Safety critical applications have realtime constraints. Security should not disable the safety function.
- Interference exists between safety and entertainment applications on the infotainment platforms.

In-vehicle security. It seems that there is still no concrete plan to change the specification of the most used automotive bus protocol, the Controller Area Network (CAN) protocol. In fact, CAN is deployed in billions of cars, industry machines and aircrafts. A specification change would impact the complete development and supply chain within the automotive industry. With CAN being the most used protocol for safety critical applications, it has been the most attractive protocol for attackers [27, 26, 37, 31, 12]. The challenge is to embed security in the CAN protocol and ensure that the safety applications are not affected by the changes.

While the CAN specification cannot be changed, using the fields of the CAN frames is a plausible alternative to embed authentication and data integrity. Today's constraints on car applications (e.g., fault tolerance times of about 100ms, message sizes of approximately 2-4 bytes, and asymmetry in the available performance in the different modules) requires the use of efficient cryptography algorithms to fulfill authentication and integrity goals. Thus, researchers have proposed to use a truncated Message Authentication Code (MAC) added in the payload of the CAN frames [50, 37]. This approach can be improved if one considers the error detection attributes of MACs. In the CAN specifications, two bytes are reserved for the Cyclic Redundancy Check (CRC) which the MAC uses. The potential of the combination of MAC and error correcting codes could be investigated as a measure to provide safety and security.

Car-to-X (a.k.a., Car2X) Security and Privacy. The success of Car2X applications depends on their penetration in the car market. Current cars will have to be upgraded for car-2o-X communications. Car2X systems will have to be compatible in terms of messages formats, communications technologies, and security, among other requirements. Moreover, applications will have to react in realtime for safety critical tasks. An example of a Car2X application is an active brake, where a car brakes based on a warning message received from an external communication. This task requires an instant brake manoeuver with a maximum delay of 250ms [21]. In 250ms the car will have to perform plausibility checks on the message content, and verify the authentication and integrity of the message. The challenges can be summarized in the following key words: Upgradeability of car systems, compatibility, realtime performance, trustworthiness, and privacy.

The fact that cybercars may need to deal with huge number of signature verifications per second enforces the use of hardware accelerators to cope with the high computation requirements for authentication and integrity verifications (e.g., of up to 4000 per second [49]). EVITA has made a

proposal for such a hardware accelerator enhanced with security features [57]. However, this module still has to be integrated in available architectures [51, 41, 6, 5] to provide a security architecture covering the security requirements of Car2X applications [22]. With the high density of Car2X communications, the common approach is to use public key infrastructure to manage the high number of required keys. An example of a public key infrastructure is proposed in [10]. One important and still open issue is the owner's privacy protection. The project Sevecom proposed the use of short term credentials (e.g., certificates), that are periodically changed by the car users [51]. Nevertheless, other layers in the communication stack may still send sensitive private information such as vehicle position.

Secure integration at the infotainment platform. Recent attacks have demonstrated that the integration surfaces at the infotainment platform (head unit) like the media players or Bluetooth are insecure [12]. The head unit typically runs three types of applications: applications with no access rights to the in-vehicle domains, applications with read access such as diagnostic tasks, and applications with read/write access such as firmware updates. Read access allows collecting privacy-sensitive data, such as location data, fuel consumption, and the vehicle identification number (VIN), which can result in creation of driver profiles. The reuse of such profiles could be interesting for businesses, e.g., insurance companies. Write access allows the control of the car which could potentially cause fatalities. The head unit may run applications with different access rights in parallel, which could lead to confused deputy attacks [25]. Present efforts to isolate the different types of applications are done in the GENIVI consortium but currently seem to rely on desktop standard approaches of virtualization [7] and do not address the unique issues of transportation systems, which is related to the safety of system users.

4. EDA CHALLENGES AND OPPORTUNITIES

Unless security is integrated within the automotive electronics and communication design flows, cybercars will be vulnerable to several nefarious attacks. The scope of the potential exploits is likely much more sophisticated than what has been demonstrated thus far in simulations or in limited practical experiments. The design, realization, and validation of complex networked embedded systems for automotive applications is already a standing challenge, even before the security demands are considered [48]. Security requirements have to be carefully recognized and implemented at each complex step of modeling, simulation, end-to-end design, and implementation of the cybercar systems.

Due to the rising complexity and scale of automotive functionalities, networks, interactions, and the supply chain, stand alone or adhoc protective solutions have a very limited effectiveness. Novel EDA methodologies and tools are required for scalable automated security modeling, integration, verification, checking, and analysis of the cybercar complex systems.

In the remainder of this section, we outline some of the EDA challenges that have to be overcome in order to realize secure cybercars and highlight the great research opportunities in this field.

4.1 Model-based automotive security

Vehicular system integration has been traditionally done based on black-box integrated subsystems along with original equipment manufacturer's high-level specifications and overall performance metrics. The design flow has to be enhanced to better model the secure cybercar system structure and interactions, possibly through refined diagrams consisting of block entities linked by interconnects and flows, which specify a topology and variables for communication among the blocks. A block may represent a logical or physical component, e.g., a hardware or software part, often either representing an abstraction of the full component description or a characterization of the component interface [44, 16, 15]. Like any other proper abstraction, neither description nor interface should contain more information than necessary to realize or connect the components. Let us more carefully investigate the design requirements and security demands for components and interfaces.

Component-models and abstractions. In order to meet the stringent time-to-market constraints, the reuse-based paradigms are a standard practice in design and implementation of complex automotive network embedded systems. Stand-alone component models are typically available and if not, they are attainable at the proper level of abstraction by the original IP owners or manufacturers. However, one has to pay a special attention to component models, especially those for the reusable ready blocks, since their functionality may not be static; their behavior has to be considered in the context of complex dynamic interactions with other system components and environmental/user variables.

Abstraction of the electronic components has traditionally been an integral part of EDA methods and tools. However, the Cyber-Physical nature of cybercar networked embedded systems requires adding a totally new dimension to the component modeling and abstraction. Essentially, there is a need to model and abstract the non-electronic components which include the mechanical parts and user inputs. In the development of such models and abstractions, challenges arise due to the inherent continuous and analog nature of the underlying components, which are far from discrete and (often) binary form of well-known digital or logical abstractions. Attacks based on exploiting the component vulnerabilities that infiltrate the electronic components, the mechanical parts, or the user inputs can be envisioned. Finding the proper level of component abstraction that also captures such potential exposures is a major challenge.

Interface-based design for security. Designers commonly make assumptions about the environment where the component will be employed, or the pairwise interactions between the components. Such descriptions can also be a part of the interface specifications. Abstracting the component security requirements brings upon yet another dimension to the already sophisticated environmental models and interactions among the underlying parts developed by disparate entities. Such secure interface-based models should support compositional refinement as well as different degrees of component abstractions. A model-based approach may profile secure control and dataflow task requirements in a graphical language amenable to graph-theoretic analysis.

According to the recent analyses of automotive security attacks in [12], virtually all exploits outlined in their practical experiments, emerged at the interface between codes

written by distinct organizations. A significant advantage of developing sound and proper component and secure interface abstractions is that they can be together used as a precursor to formal verification and correct-by-construction designs. Components and interfaces which expose protocol information about component interactions or secure protocol information can be naturally expressed in an automation-based language. A great introduction to such interface automata and interface languages are outlined in [16, 15]. They have shown that several aspects of interface models, including compatibility and refinement checking for interface interactions can be viewed in a game-theoretic framework.

4.2 Temporal models and constraints

Automotive control systems including the ECUs have traditionally been designed for real-time operations. For example, the CAN protocol implements a deterministic algorithm and assigns priorities to messages. Assuming a fault-free operation and predictable task times, the worst-case timing behavior of such a system can be estimated. However, it has been shown that especially in presence of small changes in the temporal parameters, there are points of discontinuity in the system where the increased number of preemptions adds the execution of one or more tasks to the execution time [48]. This situation is exacerbated for larger and more sophisticated interactions which in turn limit the usefulness of such predictive models.

Priority-based scheduling and security. Practical implementations of security often require a nontrivial overhead on the plaintext processing and tasks such as ciphertext decryption. The complexity would be even worse if keys must be established with external entities, such as on-the-fly Car2X communications which require public key protocols. Priority-based scheduling such as the one implemented by CAN, has very high worst-case latency and discrepancy between the best- and worst- case system timing behavior in the presence of faults or jitters. In particular, a drawback of priority-based resource scheduling is sensitivity to high priority computation or communication flows that can easily take control of the interface or the ECU in order to steal time from lower priority tasks. An intelligent attacker could use this sensitivity to attack the system timing and to violate the system's timing constraints through fault injection in high priority applications.

Additional control layers might be able to avoid the timing faults and could improve the security, but significantly add to the system overhead. Added control layers may very well lead to violations of end-to-end real-time constraints for priority-based schedules, which should be included in the system-level models discussed in Section 4.3.

Isolation of safety-critical and security sensitive tasks by time-based scheduling. Time-based schedulers such as those supported by the FlexRay enforce assignment of the communication bus at predefined time slots and are not always sensitive to system load requests [40]. For example, in FlexRay a cycle is divided into up to four segments: static, dynamic, symbol, and nit. The static part is strictly reserved for transmission of time-critical messages which have a fixed length time slot for each node. The dynamic interval is reserved for non-critical messages with more robust timing requirements. The arbitration among the less critical messages is based on priorities similar to CAN.

Time-triggered communications could allow for better isolation between security sensitive tasks and non-critical applications; they are also better suited for distributed control applications. However, time-triggered protocols are still far from providing a comprehensive security measure. For example, a denial of service attack targeted at fixed time slots could result in significant loss of bandwidth and even in the worst-case, failure in meeting realtime constraints. Thus, along with isolation, attack models as well as proactive and reactive measures to secure against attacks, must be integrated within the system timing analysis framework.

Note that isolation for security may be done by the software control layer interface standards for both priority-based and time-based scheduling policies. For example the AUTOSTAR software standard description allows isolation of timing error for one IP from the other IPs interacting with it. Since such additions may lead to timing violations, a layer in the standard is aimed at addressing performance and delay violations. In a complex control software interactions could generate timing dependencies because of scheduling conflicts, synchronization issues, or buffering. If correct timing models and abstractions are available, it is possible to set up a simulation tool which can model the behavior of tasks and their interactions in a complex environments. An interface-based design methodology could also be used for addressing the security challenges.

4.3 End-to-end secure & reliable integration

An end-to-end design of secure cybercars requires a combination of security requirements along with functional models and abstractions of the architecture and hardware platforms. Novel system-level security abstractions and analysis not only guide partitioning and separation of concerns at the design time, but also accommodate an efficient exploration of the complex design space in early design phases. Such abstractions could lead to a significant increase in efficiency of performance, reliability, delay, and implementation cost of the security solutions. We advocate the use of platform-based designs along with our suggested security flow models and simulation tools for addressing the sophisticated end-to-end system security requirements.

Platform-based design for secure architecture selection. Platform-based design [47, 46] is based upon explicitly characterized layers of abstraction and a design interface between the behavioral specifications and abstractions of possible implementation platforms. Therefore, this methodology decouples application-layer software from variations in the underlying hardware. By decoupling these two components, the same applications are permitted to run across several vehicle platforms without modifications. Once security is abstracted at the proper level, e.g., using the flow model described in Section 4.1, one would be able to use platform-based design principals and tools for system optimization. Essentially, the platform interface and the security requirements should be independent and isolated from lower-level architecture details, while simultaneously allowing design-space exploration with a good predictions of the properties for the system realization.

The design-space-exploration finds the secure system's optimal mapping into a platform candidate instance. There is a need to develop new methods and tools that can provide a measure of the appropriateness of a particular architecture solution for optimizing performance metrics while also

satisfying various performance constraints.

Iterative synthesis for security. To find a good optimization solution in the large space of possibilities, often an iterative approach is taken by the EDA community [17]. Such an iterative approach is suitable for the complex mapping between the space of security parameters/flows and the hardware platform. After a set of end-to-end design/security metrics and constraints are specified, a set of initial candidate configurations for the platform architecture are then determined. The candidates are then analyzed and compared for fitting to the design goals and security objectives. If the results are not up to expectations, alternative sets of platform architectures are iteratively evaluated.

Automated robust secure software design. Iterative synthesis and analysis results typically guide the selection of the next set of candidate architectures to evaluate. The analysis include evaluation of delay properties, timing sensitivity, faults, security, and cost. In the cybercar distributed ECU environment, software architecture and mapping can become rather complex. There is an intermediate layer between the specified function and the underlying architecture which is often implemented in software for flexibility reasons. Such tools can be automated to select the best combination of timing, security, and performance constraints.

Counterfeit parts prevention. A possible set of attacks can be launched by the counterfeit automotive parts that are prevalent in the market. The car owners' incentive in buying the fake parts is mostly driven by economics. Counterfeit car electronics not only often have various reliability problems, they could also allow for several nefarious attacks such as Trojan embedding [1, 53]. While legal measures could potentially suppress the rising problem of fake automotive components, they have not been effective in practice. This is largely because of the long and hard-to-track supply chain of fake components, the improved appearance of the cloned components, lack of sufficient reliability and security tests for the fake parts, and the black market nature of the fake parts suppliers [3, 33].

Development of methods for unclonable and secure identification of devices are very relevant for addressing these challenges [24, 36, 8, 45, 32]. Devising possible measures to disable a system when a fake component is identified, are of great interest. For cases where the exact functional description of an electronic part is unknown or a part cannot be found because of its obsoleteness, one may need to reverse engineer the functionality. Thus, research and developments in functional reverse-engineering, and in formal functional verification are important for preventing counterfeits.

4.4 Security validation and testing

System security models and exact characterization of attacks are both necessary steps for proactively or reactively protecting against pertinent vulnerabilities.

Rule checking for secure implementation correctness. There is a need for methods that define and express Security Rule Checks (SRC) based on a set of protection primitives and security protocols at the proper abstraction layer, e.g., at the control and data flow vulnerable interface levels [12]. The SRC can be used to enforce unimplemented constraints of the existing protocols which may not be always in effect, as suggested in [31]. Many widely-used strong protection protocols, including those for access control, encryption, or secure sessions, are available in a standard format. To ensure robustness of interfaces against potential adversaries, implementation of such protocols within the system's realtime constraint is necessary; this may lead to requiring hardware acceleration. Once security community spends the time and effort to develop security protocols, new SRC tools must be developed accordingly to ensure a correct implementation.

Continuous attack analysis and countermeasure development. The attacks targeted at cybercars will likely not be static and will evolve over time, as it is the case with other computer and network attacks. As researchers and practitioners devise new security primitives, rules, and protocols, adversaries could simultaneously find new holes in the system or its implementation. As the protection protocols and security methods become more advanced, the attacks will become more sophisticated. There is a need to continuously and dynamically monitor for potential vulnerabilities and attacks. Once instances of attacks are observed, the corresponding countermeasures should be implemented within the system. Software patches and anti-virus software should be continually updated to limit the spread of any exploits. Online tests for detection of possible exploits should also be implemented and enforced in the cybercar systems.

Counterfeit detection. Since counterfeits provide physical access to the system, malicious fake devices could potentially launch efficient attacks. They may be Trojans that spy information to the outside world or disrupt the system's functionality on a trigger event. Even when the counterfeit parts are not intentionally malicious, they introduce a high risk to the system reliability [33]. This is because the counterfeit components are often lower quality grades, or recycled old ICs. Therefore, the system designers must automatically embed means for testing and detecting the discrepancy and unreliability of the system components, and for monitoring/detecting the Trojan components.

Development of cybercar security benchmarks. Like other areas of EDA and testing, it is necessary to create benchmarks and platforms in order to evaluate and compare the competing methodologies. A flurry of research activities are being directed towards addressing the known and potential vulnerabilities of cybercar systems. There is a need to understand the effectiveness and limitations of each method. This will not be only used as an evaluation tool, but also would help in standardization of the best methodologies and tools. Effective security benchmark development requires outlining the taxonomy and details of the attacks, as well as research and experiments which realistic but challenging hard-to-address attack instances.

5. CONCLUSION

Tight integration of networked computation and physical components in safety-critical modern automotive environments and applications (cybercars) makes them exceptionally vulnerable to security attacks, as confirmed by a number of recent studies. The evolving nature of the complexity of cybercar systems and the severity of possible attacks suggest integration of security within the embedded system design flow. This perspective article highlights the importance of developing novel holistic and systematic EDA methodologies and tools that simultaneously ensure a robust and secure cybercar design flow. We discussed the challenges and opportunities in realizing our proposed vision.

APPENDIX

A. RELATED LITERATURE

Modern vehicular technology is typically comprised of several intercommunicating electronic and software components that enable ever-increasing flexibility, efficiency, safety, and a myriad of new and exciting functionalities [14]. While automotive systems, standards, functions, and components are almost always devised to satisfy strict safety and reliability constraints, requirements for security and protection of the pertinent components and functions have not yet been implemented. Therefore, concerns over the potential risks and vulnerabilities of this complex networked environment were expressed at a high level [59, 39, 58, 34, 35, 54]. Such worries have been exacerbated by the emergence of newer car-to-X technologies [42, 29].

The EVITA project took the first fundamental step towards addressing the rising concerns about the security of modern cars. The project identified a list of automotive use cases with a security impact that could be potentially misused [21]. It also developed a security and trust model for modern vehicles along with attack scenarios [22]. To address identified vulnerabilities, a set of high level requirements along with concrete technical recommendations have been proposed [23]. Architecture solutions that could address the identified vulnerabilities were subsequently suggested and prototyped in hardware.

A majority of the reported attacks have been targeted at the Control Area Network (CAN) protocol. Attacks on simulation models of CAN were suggested in [26, 38]; the first implementation on a real car was demonstrated in [27], where the researchers performed several practical tests on the CAN network and demonstrated their vulnerabilities. The demonstrated attacks included controlling the windows, lights, and airbag systems. Later, they discussed CAN privacy violation issues [28].

The authors in [43] performed an analysis of the Wireless Tire Pressure System (TPMS) in a modern automobile. They outlined methods to manipulate drivers by spoofing the faulty tire pressure measurements. The tire pressure values are typically sent from the tire pressure wireless sensor to the ECU managing the TPMS data. The authors were able to send wrong values to stop the functionality of the ECU managing the TPSM data. Note that our focus in this work is on the intrusions at the network level that are orthogonal to the practical demonstration of attacks on single-device vehicle access control mechanisms such as those targeting the keyless entry system [19] and vehicle immobilizers [11].

A more comprehensive practical security analysis of the CAN vulnerabilities was later demonstrated in [31], where cars' components were tested in isolation in a lab, in controlled settings, and in live road tests. Their attacks confirm that an attacker infiltrating any ECU in the CAN network could circumvent a large array of of automotive functions while ignoring the driver inputs. The attacks included safety critical tasks such as break disabling, engine halting, and light control. Other less critical but vulnerable modules were heating and cooling, infotainment, and instrument panels. Their findings show the extent of potential damages from the existing security holes, ease of attacks, weakness of contemporary vehicular access control, and the ability to delete the traces of infiltration by the attackers. Lastly, they explore considerations and future possible high-level directions for addressing some of the known vulnerabilities.

With the exception of the wireless access experiment in [43], most of the attacks described thus far assume the adversary has physical access to the car's internal network. More recent work in [12] provided a systematic synthesis of possible attack vectors at three modalities: indirect physical access, short-range wireless access, and long-range wireless access. For each modality, the authors reported the practicality of exploitable vulnerabilities, allowing unauthorized control without the need for physical access. They further demonstrated a number of post-compromise control channels that could act like a remotely controllable Trojan. Capabilities of theft and surveillance were also shown. A set of high-level recommendations for raising the security bar were subsequently suggested. We believe that these efforts are a precursor to future research and developments as we are still far from fully secure and protected cybercars.

B. REFERENCES

[1] Defense science board (DSB) study on high performance microchip supply. http://www.acq.osd.mil/dsb/reports/2005-02-hpms_report_final.pdf.

[2] ASAM: Association for standardisation of automation and measuring systems. http://www.asam.net.

[3] Defense industrial base assessment: Counterfeit electronics. Technical report, U.S. Department Of Commerce, Bureau Of Industry And Security, Office Of Technology Evaluation, 2010.

[4] AUTOSAR automotive open software architecture specification. http://www.autosar.org, 2012.

[5] CALM communications access for land mobiles. http://calm.its-standards.eu/, 2012.

[6] ETSI TC ITS standards european telecommunications standards institute. http://www.etsi.org, 2012.

[7] G. Alliance. Genivi. `http://www.genivi.org/`, 2009.

[8] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, and C. Wachsmann. A formalization of the security features of physical functions. In *IEEE Symposium on Security and Privacy*, pages 397–412, 2011.

[9] A. Barisani and D. Bianco. Unusual car navigation tricks. In *CanSecWest*, April 2007.

[10] N. Bissmeyer, H. Stuebing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc. A generic Public Key Infrastructure for securing Car-to-X Communication. In *18th ITS World Congress, Orlando, USA*, Oct. 2011.

[11] S. C. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo. Security analysis of a cryptographically-enabled RFID device. In *USENIX Security Symposium*, pages 1–16, 2005.

[12] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, 2011.

[13] C. C. Consortium. Mirror link. `http://www.terminalmode.org/`, 2010.

[14] J. Cook, I. Kolmanovsky, D. McNamara, E. Nelson, and K. Prasad. Control, computing and

communications: Technologies for the twenty-first century model T. *Proceedings of the IEEE*, 95(2):334 –355, 2007.

[15] L. de Alfaro and T. Henzinger. Interface theories for component-based design. In T. Henzinger and C. Kirsch, editors, *Embedded Software*, volume 2211 of *Lecture Notes in Computer Science*, pages 148–165. Springer Berlin / Heidelberg, 2001.

[16] L. de Alfaro and T. Henzinger. Interface-based design. In M. Broy, J. Grünbauer, D. Harel, and T. Hoare, editors, *Engineering Theories of Software Intensive Systems*, volume 195 of *NATO Science Series*, pages 83–104. Springer Netherlands, 2005.

[17] G. De Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Higher Education, 1st edition, 1994.

[18] F. Dressler, F. Kargl, J. Ott, O. Tonguz, and L. Wischhof. Research challenges in intervehicular communication: lessons of the 2010 dagstuhl seminar. *Communications Magazine, IEEE*, 49(5):158 –164, 2011.

[19] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani. On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In *CRYPTO*, pages 203–220, 2008.

[20] EVITA Consortium. The EVITA project: E-safety vehicle intrusion protected applications, 2011. `http://www.evita-project.org`.

[21] The EVITA project Deliverable 2.1, http://evita-project.org/deliverables.html. *Specification and evaluation of e-security relevant use cases*, 2009.

[22] The EVITA project Deliverable 2.3, http://evita-project.org/deliverables.html. *Security requirements for automotive on-board networks based on dark-side scenarios*, 2009.

[23] The EVITA project Deliverable 3.1, http://evita-project.org/deliverables.html. *Security and trust model*, 2009.

[24] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *ACM Computer and Communications Security Conference (CCS)*, pages 148–160, 2002.

[25] N. Hardy. The confused deputy (or why capabilities might have been invented). *Operating Systems Review*, 22(4):36–38, 1988.

[26] T. Hoppe and J. Dittman. Sniffing/replay attacks on CAN buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In *Workshop on Embedded Systems Security (WESS)*, 2007.

[27] T. Hoppe, S. Kiltz, and J. Dittmann. Security threats to automotive CAN networks - practical examples and selected short-term countermeasures. In *Computer Safety, Reliability, and Security (SAFECOMP)*, pages 235–248, 2008.

[28] T. Hoppe, S. Kiltz, and J. Dittmann. Automotive IT-security as a challenge: Basic attacks from the black box perspective on the example of privacy threats. In *Computer Safety, Reliability, and Security (SAFECOMP)*, pages 145–158, 2009.

[29] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications Magazine*, 46(11):110–118, 2008.

[30] P. Kleberger, T. Olovsson, and E. Jonsson. Security aspects of the in-vehicle network in the connected car. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 528 –533, 2011.

[31] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy (S& P)*, pages 447–462, 2010.

[32] F. Koushanfar. Provably secure active ic metering techniques for piracy avoidance and digital rights management. *IEEE Transactions on Information Forensics and Security*, 7(1):51–63, 2012.

[33] F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, M. Sale, P. Song, and M. Potkonjak. Can EDA combat the rise of electronic counterfeiting? In *Design Automation Conference (DAC)*, 2012.

[34] A. Lang, J. Dittmann, S. Kiltz, and T. Hoppe. Future perspectives: The car and its IP-address - a potential safety and security risk assessment. In *International Conference Computer Safety, Reliability, and Security (SAFECOMP)*, pages 40–53, 2007.

[35] U. E. Larson and D. K. Nilsson. Securing vehicles against cyber attacks. In *Cyber Security and Information Intelligence Research Workshop (CSIIRW)*, pages 30:1–30:3, 2008.

[36] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Lightweight secure PUFs. In *International Conference on Computer-Aided Design (ICCAD)*, pages 670–673, 2008.

[37] D. K. Nilsson, U. Larson, and E. Jonsson. Efficient in-vehicle delayed data authentication based on compound message authentication codes. In *VTC Fall'08*, pages 1–5, 2008.

[38] D. K. Nilsson and U. E. Larson. Simulated attacks on CAN buses: vehicle virus. In *IASTED International Conference on Communication Systems and Networks (AsiaCSN)*, pages 66–72, 2008.

[39] C. Paar, A. Weimerskirch, and M. Wolf. Security in automotive bus systems. In *Embedded Security in Cars Workshop*, 2004.

[40] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei. Timing analysis of the FlexRay communication protocol. In *Euromicro Conference on Real-Time Systems*, pages 203–216, 2006.

[41] PRESERVE Consortium. Preparing secure vehicle-to-x communication systems, 2011. `http://www.preserve-project.eu`.

[42] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing vehicular communications. *IEEE Wireless Communications,*, 13(5):8–15, 2006.

[43] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless

networks: a tire pressure monitoring system case study. In *USENIX Security Symposium*, 2010.

[44] J. Rowson and A. Sangiovanni-Vincentelli. Interface-based design. In *Design Automation Conference (DAC)*, pages 178–183, 1997.

[45] U. Ruhrmair, S. Devadas, and F. Koushanfar. *Security based on Physical Unclonability and Disorder, Book Chapter in 'Introduction to Hardware Security and Trust'*. Springer, 2011.

[46] A. L. Sangiovanni-Vincentelli. Defining platform-based design. *EE Times*, Feb 2007.

[47] A. L. Sangiovanni-Vincentelli. Quo Vadis, SLD? reasoning about the trends and challenges of system level design. *Proceedings of the IEEE*, 95(3):467 –506, 2007.

[48] A. L. Sangiovanni-Vincentelli and M. Di Natale. Embedded system design for automotive applications. *IEEE Computer*, 40(10):42–51, 2007.

[49] E. Schoch and F. Kargl. On the efficiency of secure beaconing in vanets. In *Proceedings of the third ACM conference on Wireless network security (WiSec)*, pages 111–116, 2010.

[50] H. Schweppe, M. Idrees, Y. Roudier, B. Weyl, R. El Khayari, O. Henniger, D. Scheuermann, G. Pedroza, L. Apvrille, H. Seudié, H. Platzdasch, and M. Sall. Secure on-board protocols specification. Technical Report Deliverable D3.3, EVITA Project, 2010.

[51] SEVECOM Consortium. Secure vehicular communication, 2008. `http://www.sevecom.org`.

[52] SIMTD Consortium. Sichere intelligente mobilität testfeld deutschland, 2011.

[53] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE Design & Test of Computers*, 27(1):10–25, 2010.

[54] P. R. Thorn and C. A. MacCarley. A spy under the hood: Controlling risk and automotive EDR. In *Risk Management*, 2008.

[55] C. to Car Communication Consortium. Car to Car Communication Consortium. `http://www.car-to-car.org/`, 2006.

[56] VDAT. Der deutsche tuningmarkt. `http://www.vdat.org/tuningmarkt_deutschland.php/`, 2010.

[57] B. Weyl, M. Wolf, F. Zweers, T. Gendrullis, M. Idrees, Y. Roudier, H. Schweppe, H. Platzdasch, R. El Khayari, O. Henniger, D. Scheuermann, A. Fuchs, L. Apvrille, G. Pedroza, H. Seudié, J. Shokrollahi, and A. Keil. Secure on-board architecture specification. Technical Report Deliverable D3.2, EVITA Project, 2010.

[58] M. Wolf, A. Weimerskirch, and T. J. Wollinger. State of the art: Embedding security in vehicles. *EURASIP Journal of Embedded Systems*, 2007.

[59] Y. Zhao. Telematics: safe and fun driving. *IEEE Intelligent Systems*, 17(1):10–14, jan/feb 2002.