Security Requirements Engineering in the Automotive Domain: On Specification Procedures and Implementational Aspects

Marko Wolf and Christof Paar

Abstract: It is not necessary to always present the terrorist attacker that encroaches into another's ABS as the dramatic example to alert the need for vehicular IT security. It may suffice to imagine some very simple encroachments on in-vehicle communication integrity or on functional availability that could cause a suddenly uncontrolled spattering windshield washer or a malfunctioning door look system that—in the wrong situation—actually can suffice to threaten life [Ban03]. Although most vehicular applications are developed to face (random) technical failures, they almost never consider a human malicious encroachment. Hence, many vehicular IT systems are susceptible to security issues and hence, can quickly become also safety-critical.

This work describes how to identify the individual security objectives of all entities involved in a typical vehicular IT application. It describes how to deduce the corresponding security requirements that fulfill the security objectives identified before. Finally, this work indicates some helpful vantages and several constraints characteristical when establishing IT security in the automotive domain.

1 Motivation

Security-critical vehicular applications means applications that require, in addition to measures against (random) technical failures (i.e., safety measures), measures that thwart malicious encroachments, which enable unauthorized access or manipulations, or simply affect the availability of certain vehicular functionality or information. So far, there only existed niche applications in the automotive domain (e.g., immobilizers, digital tachograph) that particularly rely on security technologies. However, the situation has changed dramatically. More and more vehicular systems need security functionalities in order to reliably provide driving and road safety, and to protect the revenues, the liability, and the reputation of manufacturers and suppliers as well as to protect the various interests of the vehicle owner. Secure software updates for electronic control units (ECU), chip tuning prevention, mileage protection, and protection against counterfeit vehicular components are only some current examples for the need for vehicular IT security measures. Future vehicles will become even more dependent on IT security due to the following developments.

- Vehicles are increasingly controlled by electronics, where again software more and more determines the functionality and hence becomes the dominant factor [Fri04].
- Vehicle electronics are increasingly linked in (unprotected) internal networks that again will be linked with open unprotected external networks [PWW04].

- Vehicle hardware and software becomes more and more standardized [Bro06] making potential attacks more scalable and increase potential payoff.[1]
- Vehicles have to fulfill many new legislative requirements (e.g., eCall [Eur07]).
- Vehicles will be involved in various new business models (e.g., aftermarket, pay-per-use, or location-based applications [RG06]).
- Vehicles will communicate wirelessly with their environment to enable new safety (e.g., electronic traffic signals), comfort and business applications [RPH06].

In fact, most vehicular IT systems are susceptible to security issues and hence, can quickly become also safety-critical. Even, at the first glance, fully non-critical vehicular applications can have serious impacts on driving safety. It suffices to imagine some (malicious) encroachments that could cause for instance a suddenly uncontrolled spattering windshield washer (without the wipers going on), an uncontrolled acting electronic park brake, or the car radio going abruptly on full blast—all possibly in a critical traffic situation. Even a simple malfunction of the electronic door look system can already become life threatening, if it prevents leaving the car in a critical situation [Ban03]. Although most vehicular applications are developed to face technical failures (e.g., by verifying checksums), they almost never consider a systematic human attacker that uses a certain functionality indeed syntactically correct, but in bad faith. Especially, since attacks that for instance simply target the availability of a certain vehicular functionality often can be mounted very easily. However, in contrast to a jet pilot for instance, who can take strong security and safety precautions for granted, an ordinary car driver usually has not even enough time nor sufficient options to react effectively in case of a (malicious) failure, which, particularly in most traffic situations, can very quickly cause serious consequences [Dri02].

Nevertheless, currently, most vehicular IT systems are not protected against such malicious encroachments. The reason that hitherto existing vehicular IT systems seldom included any security functionality was, that there was only little potential payoff and thus only little incentive for malicious manipulations [MZ05]. Secondly, IT security measures first of all cause technical and organizational costs without directly visible (i.e., promotional) benefits for manufacturers or customers. Finally, security often tends to be an afterthought in many IT systems[2], because achieving the core functionality first, is a widely used approach even when designing a critical vehicular IT system. In fact, most vehicular electronics were rapidly growing bottom up, from simple, isolated, and dedicated microcontrollers, which became highly complex, interconnected, and interactive distributed systems over the last 30 years [Bro06]. Hence, a top down design approach based on systematic software and security-engineering principles usually was never applied. However, an ab initio awareness also for malicious encroachments and an ab initio integration of appropriate IT security measures is inevitable when developing a critical vehicular IT system. This makes proper security requirements engineering an inherent part in the development of most vehicular IT applications.

[1] Nonetheless, ongoing standardization certainly provides also some very valuable and indispensable characteristics that increase vehicular safety and security (cf. Section 4.1).

[2] As can be seen for instance by the development of some operating systems or several Internet applications, implementing IT security afterwards is normally doomed to failure.

Main Contribution. This work describes how to identify the individual security objectives of all entities involved in a typical vehicular IT application. It then describes how to deduce the corresponding security requirements that fulfill the security objectives identified before. Finally, this work indicates some helpful vantages and several constraints characteristically when establishing IT security in the automotive domain.

Outline. This work has started with a motivating section, followed by a section that depicts how to identify the individual security objectives of all entities involved in a typical vehicular IT application, in Section 1 and Section 2 respectively. Section 3 then describes how to deduce the corresponding security requirements that fulfill the security objectives identified before. Section 4 finally indicates helpful vantages and several constraints characteristically when implementing security requirements in the automotive domain, before this work closes with a conclusion.

2 Security Objectives Analysis

To guarantee road safety and operational reliability of vehicles and to sufficiently protect automotive business models, legacy and comfort applications that are based on the security of the vehicular platform, the following overall security objectives (SO) are reasonable. Note that these security objectives are additional objectives, in addition to other mandatory, but not security related objectives such as usability or scalability (cf. Section 4.2).

Authenticity. The origin of a vehicular information (e.g., message, program, or data) or a vehicular component (e.g., hardware, firmware) must be verifiable. Particularly, unauthorized cloning of a protected vehicular hardware component must be infeasible or at least detectable as falsified.

Confidentiality. Unauthorized access to protected or private resources (e.g., trade secrets, personal or proprietary data) must be infeasible.

Integrity. Unauthorized modifications of a vehicular information or a vehicular component must be infeasible or at least detectable either internally or by a regular and non-predictable controlling entity. Particularly, any kinds of replay attacks must be infeasible.

Policy enforcement. Circumvention of effective security policies[3], which all legitimately involved parties have accepted, must be infeasible.

Availability. Authorized entities (e.g., hardware modules, software processes, users) must have proper and timely access to their dedicated data and services.

A further, but overall security related objective is *privacy* such that the usage of any vehicular application must not endanger the privacy policies of the respective driver/owner and thus should prevent harm, embarrassment, inconvenience, or unfairness to any party whose data is processed [Shi00]. However, privacy protection is mainly based on the overall design of a vehicular application, the information actually involved, the specific context, as

[3] Such a security policy could be, for instance, an access or execution control policy or a policy that certain hardware components have to be verified for OEM's legitimacy before installation.

	Vehicle Owner	ECU IP Owner	Flashing Operator	Vehicle OEM
Authenticity	■	■	□	■
Confidentiality	□	■	□	□
Integrity	■	■	■	■
Policy Enforcement	□	□	□	■
Availability	■	□	■	■

Table 1: Exemplary mandatory (■) and optional (□) security objectives on flash data of a safety-critical ECU depending on the corresponding entity.

well as general organizational and legal measures. Security objectives such confidentiality, however, can be necessary to realize privacy protection. Moreover, the concrete security objectives generally depend on the respective application and may be different for each information/resource and each entity involved. As shown in Table 1, in the exemplary case of a safety-critical ECU software update, the only entity that may be actually interested in information confidentiality could be the ECU IP owner in order to protect its intellectual property (IP). However—in contrast to the vehicle owner, the flashing operator, and the OEM—the IP owner may be less interested in the OEM's policy enforcement or local availability, whereas each of the involved entities is interested in the integrity of the flash-memory data.

In order to derive the respective security objectives for a security-critical application, the following procedure is proposed. Each of following steps is further exemplarily illustrated by the security objective analysis of a vehicular event data recorder (EDR) [4] application. Anyhow, further readings on this subject are helpful and can be found amongst others in [And01, ISO04a, ISO04b, ISO05b].

1. Identify all involved, potentially critical data.
2. Identify all involved entities.
3. Identify the security objectives of each involved entity on each of the identified data.
4. Merge the security objectives of all involved entities for each data.

After choosing a security-critical application[5] for SO analysis, all potentially critical data involved are identified. As pointed out by the example of a suddenly uncontrolled spattering windshield washer, even data that seem non-critical at the first glance should be included at the begin[6] of the analysis. In case of the EDR example, such data could be

[4] The event data recorder is an at least tamper-evident device that is already installed in some of today's automobiles and trucks to record vehicle's recent driving-related activities. Similar to a "black box" known from airplanes, information collected from such a device can help for instance to clarify the circumstances of an accident.

[5] Note, as exemplarily shown in the motivating section, even vehicular applications that seem non-critical at the first glance may be susceptible to security issues that can lead to serious impacts on driving safety.

[6] If it turns out afterwards that for certain data none of the involved entities has any security objective on, they are automatically ceased from further analysis.

several vehicle sensor data (e.g., current speed or acceleration), vehicle control data (e.g., current steering or pedal positions), the current time, and the current (absolute) position. Now, all involved entities will be identified, that means all concrete or abstract roles and parties, which may have or would like to have access to the data identified before. In the EDR example, this could be the actual driver, the owner, the garage personnel, a supplier, the OEM, several authorities (e.g., police, justice), or for instance an insurance company. Having figured out all involved data and relevant entities, the security objectives of each entity on each of the involved data are identified. In the EDR example, the driver could demand for integrity on the clock data, whereas certain authorities would moreover require authenticity and availability of the clock signal. Otherwise, in contrast to a potentially involved insurance company, the OEM could have no security objectives on actual position and time, but would require authenticity, integrity, and availability for all involved sensor and control data. Finally, to meet the security objectives of all involved entities, the security objectives on each data are merged together. In case of the EDR example, this could result in the combined security objectives authenticity, integrity, and availability on the processed clock data.

3 Security Requirements Engineering

Security requirements (SR) are the actual measures or functionalities needed to fulfill the security objectives (SO) identified before. The security requirements, in turn, heavily depend on the actual security environment, that is, where and how the respective application is deployed, which assumptions can be made, which security policies are relevant, and particularly, which attack potentials arise. Having identified the effective security objectives, the following procedure is proposed. To illustrate each step of the security requirements engineering as well, the EDR example application from the previous section will be used further on. However, again further readings are helpful and can be found amongst others in [And01, ISO05a, ISO04b, ISO07, MN03, Rus01].

1. Identify all vehicular components that handle data covered by security objectives.
2. Identify the effective security environment; concretely, identify effective assumptions, all relevant security policies (if any) and all potential threats (i.e., attacker model, attack vectors[7] or so called “abuse cases”).
3. Estimate the respective attack potential [CCD07] and derive appropriate security requirements to meet the security objectives for the concrete security environment.

First, all vehicular components that handle (e.g., read, write, modify, or transmit) data, which are covered by security objectives, are identified. This includes components of the actual application as well as supplementary components, which the application shares with other applications (e.g., certain sensors). In case of the EDR example, the involved

[7] An attack vector is a path, procedures, or means by which a malicious entity can gain a malicious outcome without interpretation of their feasibility.

components could be several sensor and control ECUs, the clock component(s), the internal vehicular communication system (e.g., the CAN bus, FlexRay), and the actual data recorder component. To identify the effective security environment, first potential attackers together with their respective possible intentions have to be identified. This means a detailed analysis about who will be interested in accessing, destroying or manipulating data (or functionality), that is, who will be interested in circumventing a given security objective and how a potential attacker can gain a malicious outcome (cf. [WWW07]). In the EDR example, a potential attacker could be the current driver trying to manipulate a record, which could be interpreted to his disadvantage. Another attacker could be the OEM or an insurance company, trying to gain more information than they are entitled to (e.g., in order to detect a possible driver's malpractice) that would clearly affect driver's privacy objective. A further part of the security environment analysis is to identify all relevant security policies and effective assumptions. An assumption for the EDR example could be that the actual recorder component is sufficiently physically secure[8]. Effective assumptions also consider feasible access perimeters of potential attackers (e.g., only logical or also physical access), their technical, financial, and knowledge resources and the potential feasibility of an offline attack[9]. In case of the EDR example, the driver may have physical and logical access to the internal vehicular bus system that communicates security-critical data, whereas an insurance company would have only indirect, logical access to security-critical data, but may be able to mount an offline attack.

Having identified all involved components, the effective security environment, and all potential threats, the respective attack potentials have to be estimated accordingly. Based on a feasibility analysis of all identified attacks [CCD07], appropriate security requirements (that could also include a specific assurance level) can be derived that fulfill the defined security objectives. In the exemplary EDR application, such a resulting security requirement could be *component authentication* that allows the recorder component to verify the identity (and thus the associated assumptions on correctness and trustworthiness) of the clock component, which has to withstand an attacker with basic attack potential. Another security requirement could be *secure bus communication* to prevent manipulations of the clock signal during the transmission between the clock component and the actual recorder component, which has to withstand an attacker with moderate attack potential. In the following, some further security requirements, which are typically deduced for security-critical applications in the automotive domain, are given.

Access control mechanisms (e.g., discretionary access control and/or mandatory access control [DoD85]) prevent unauthorized access to restricted vehicular data or restricted vehicular resources (e.g., networks, computing power). Access rules to restricted data and resources are defined in the corresponding security policy derived during the overall security requirements engineering process, which determines the access rights for each authorized entity (cf. Table 2).

[8] However, such an assumption should be in turn the result of another independent security analysis.

[9] Attacks on vehicular IT systems are normally *offline attacks*, where an attacker has virtually unlimited time and virtually unlimited trials to successfully mount an attack. Hence, the attacker can calmly mount almost any feasible attack without having to fear to be detected, back traced, or locked out.

	Actuator	Driver	Garage	GPS	OEM	Police	Sensor
Motion sensor data	☐	r	☐	☐	r	r	w
Steering data	w	r	☐	☐	r	r	☐
Clock & position	☐	r	☐	w	☐	r	☐
Maintenance data	☐	r	rw	☐	r	r	☐

Table 2: Exemplary discretionary access control matrix for data processed by the data recorder component of an EDR application (r: read access, w: write access, ☐: no access).

Component identification & authentication provides verification of the component identifier and the component authenticity. This allows protection of original and legitimate components against counterfeits, thefts, or unauthorized installations by binding critical components to the corresponding vehicle.

Identity Management ensures secure and privacy-preserving creation, assignment, description, management, and deletion of identifiers for users, hardware, software, or processes.

Secure audit protects monitoring of certain vehicular information, actions, and events upon acceptance of all legitimately involved parties. A secure audit then ensures authenticity, availability, and integrity of records.

Secure communication provides confidentiality, integrity, and non-repudiation of the communicated information. It further enables the verification of the authenticity of the communication endpoints (*secure channel*) and could additionally also enable the verification of the configuration of the communication endpoints (*trusted channel*) in order to determine its trustworthiness [AES+07].

Secure initialization ensures the integrity (authenticity, non-repudiation, and freshness) of a vehicular (sub-)system during start of operation as result of a foregoing deactivation or as part of its initial installation.

Secure storage provides confidentiality, integrity, freshness, and availability of information persistently stored.

Secure provision of sensor data provides availability, authenticity, and integrity of information provided by a sensor.

Strong isolation ensures that subsystems, components, and even individual applications can communicate only via strictly controlled communication channels[10] such that it is impossible to access (i.e., data, functionality) or even affect (e.g., performance) each other without proper authorization.

User identification & authentication provides verification of the user identifier and the user authenticity. This prevents unauthorized access to and user-based access control for restricted vehicular data or restricted vehicular resources.

[10] Strictly controlled communication channels can be provided by a very small and hence verifiable hardware-based and/or software-based separation mechanism such as ARM's TrustZone technology [WFM+07] or several virtualization technologies [MLO97, SJV+05].

4 Implementational Aspects of Vehicular Security Requirements

By being located between the world of general-purpose computers and the strict embedded world (i.e., cellular phones or smartcards), a vehicular IT environment provides some characteristical vantages, but is inherently also subject to some characteristical technical and non-technical constraints, which may considerably affect, restrict, or even prevent the implementation of certain security requirements. Thus, the realization of a vehicular security requirement inherently involves a detailed analysis of effective ancillary conditions.

4.1 Characteristical Advantages

A vehicular IT environment provides some characteristical advantages, which can ease the realization of certain security requirements considerably. Some of them are briefly described in the following.

Feasibility of Updates. Even though software and particularly hardware updates are quite restricted in extent and frequency, they are at least feasible in a limited manner. Particularly, since automotive OEMs are usually liable for up to 20 years if critical faults or vulnerabilities have been identified. Thus, they are heavily interested in timely applying necessary (security) updates. Vehicle owners are usually interested in an up-to-date status just as well to ensure reliability and property retention of their vehicle.

Periodic Inspections. Vehicles are usually subject to periodic predictable (e.g., by a technical inspection authority) and non-predictable (e.g., by the police) inspections by an official control entity so that possible (successful or even attempted) attacks often can be detected afterwards and subsequently lead to non-technical (legal) actions.

Moving Target. Since a vehicle usually continuously changes its physical location (i.e., in contrast to a general-purpose computer system), at least an external attacker has comparatively limited time and limited trials to successfully attack and encroach a vehicle.

Physical Protection. Even though many security-critical vehicular applications have to deal with attackers, which may have also physical access to crucial components, vehicles usually provide, up to a certain extent, some physical protection that, according to the attacker's access perimeters, may at least complicate many attacks considerably.

Sufficient Energy and Space. In contrast to strictly embedded devices (e.g., mobile devices or smartcards); there are somewhat weaker restrictions on power consumption and devices' size and weight, which allow implementing somewhat more sophisticated and costly (with regard to size and power) security functionalities.

Ongoing Standardization Efforts. Ongoing standardization of involved (security) hardware, software, interfaces and protocols, indeed makes potential attacks more scalable (cf. Section 1), but, on the other hand, also allows regular careful verifications for correctness and immunity, makes security less costly and reduces the application of error-prone proprietary security solutions.

4.2 Characteristical Constraints

Despite some helpful vantages (cf. Section 4.1), vehicular IT environments also implicate some characteristical technical and non-technical constraints, which may considerably affect, restrict, or even prevent the realization of certain security requirements. Some of them that have to be carefully considered are briefly described in the following.

4.2.1 Technical Constraints

In the following several typical technical constraints, which may affect the realization of various vehicular security requirements, are described.

Limited Computing Resources. Computing resources of vehicular components are, in comparison to general-purpose computer systems, rather limited due to the typically strong cost (weight and energy) requirements. Nevertheless, automotive applications are often required to provide (hard) real-time capabilities. This leads to severe restrictions on complexity, memory size, and runtime efficiency for automotive security implementations that moreover often have to cope with lots of specific architectural restrictions, which often means costly, low-level, and hardware-specific implementations. However, highly customized code increases maintenance and decreases its reusability.

Physically Challenging Environment. Vehicular IT systems are often subject to specific physical constraints such as high variations in temperature, moisture, or particular mechanical loads. They have to cope with these conditions usually over a product life cycle of up to 20 years in which only minimal maintenance efforts are acceptable.

Limited External Communication Resources. A vehicular IT system usually has only very limited communication resources to, for instance, exchange cryptographic keys, update certificates, or to adapt revocation lists. Thus, virtually all vehicular security functionality has to work properly even with an external communication functionality severely limited in capacity and frequency.

Limited Update Feasibility. The application of frequent security updates is, in comparison to general-purpose computer systems, rather limited. Even if software-based security issues could be fixed almost automatically using available external communication resources, security vulnerabilities in hardware components usually would require costly and irritating recall procedures that means the vehicle owner has to locate for an authorized garage to fix up the respective component. Moreover, once millions of vehicles are sold, over many years, all over the world, it is almost infeasible to fix a critical vulnerability afterwards on all vehicles in use.

Limited Complexity for User Interaction. Since typical computer users can mostly employ ergonomic input and output devices to accomplish for instance user authentication or certain security settings, users within the automotive environment are restricted to only little ergonomically designed man-machine interfaces (MMI). To demand only a minimum of user interactions, virtually all vehicular applications are required to run almost completely

autonomous. However, if user interactions are inevitable, they have to be practicable as simple, flexible and little cumbersome as possible.

Increasing System Complexity and Diversification. The high complexity as well as the multifunctional, diversified, and distributional nature of current vehicular electronics is contrary to the security principle of economy, such that corresponding security measures can be "designed to be as simple as possible, so that the mechanism can be correctly implemented and so that it can be verified that the operation of the mechanism enforces the containing system's security policy" [Shi00]. Current vehicles already provide up to 2000 individual software-based functions, which inherently generate several ten thousand pages of technical documentation [Bro06]. Future vehicles will further increase application diversity. This clearly complicates a comprehensive realization, and in particular a complete verification, of security measures that are able to reliably protect such a complex IT system. However, ongoing standardization efforts (cf. Section 4.1) will at least help to counteract an excessively rampant proliferation of proprietary functionalities.

Distributed Architecture. Vehicular IT systems and the involved components (e.g., sensors, busses, or controllers) usually are very heterogeneous and widely distributed over the whole automobile. This creates many vulnerable points to successfully mount an attack, and hence considerably complicates a holistic protection of security-critical applications.

4.2.2 Non-technical Constraints

Beyond the technical constraints, vehicular IT systems are also subject to several typical historical, organizational, and legal constraints, which may considerably affect or complicate the realization of various security requirements.

Unfamiliar Technology. IT security was too long an only little-noticed subject in the development of vehicular applications and services. With the rapid introduction of more and more software-driven vehicular components, IT security suddenly becomes an essential technology for many automotive developments. However, the characteristical traditional structures in the automotive industries, with their multitude of proprietary and isolated developments, make it rather difficult to properly implement IT security, which particularly demands holistic, top-down approaches. Strictly speaking, current vehicular infrastructures were never intended to also implement IT security measures. Thus, there hardly exist any up-to-date standards, rules, or specifications regarding vehicular IT security, while current standardization and specification efforts can hardly keep up with the fast ongoing development of vehicular electronics. Furthermore, automotive engineers normally are mechanical engineers or electrical engineers without any special training in security. Given the tricky pitfalls often inherent in security design, this is often a major real-world hurdle[11]. Hence, the automotive industry is missing experts in IT security, which are already rare in most other industries with a need for security experts (e.g., aircraft industry).

Isolated Subsystem Development. For historical reasons, most vehicular subsystems are

[11] As it for instance happened with the DVD video copy protection [Pat99], an ill-conceived security design is quickly broken.

developed and produced completely independent from the corresponding OEM by individual independent suppliers, which in turn often serve several different OEMs at the same time. However, current software-driven development considerably increases mutual relation and interaction of vehicular subsystems with each other, which makes independent and autonomous security measures often impractical or even impossible. Holistic security measures generally require integrated top-level approaches; that is, OEMs have to synchronize the developments of their suppliers to manage dependencies and prevent vulnerable interactions.

Multitude of Involved Parties. The current multi-tier vehicular manufacturing process (OEM and possibly several layers of suppliers) can complicate the realization of security solutions considerably. It will be difficult, for instance, to decide who is in charge for the overall security design and, in particular, who has control over the corresponding cryptographic secrets. A potential public key and certificates infrastructure (PKI) as well as the overall security management requires further complex and costly organizational structures and involves even more different parties (e.g., manufacturer, supplier, OEM, garage personnel, content provider, etc.) with very different security understanding.

Additional Costs with Little Promotional Benefits. In contrast to the implementation of novel functionalities, implementing IT security first of all creates costs without apparent (i.e., promotional) benefit for suppliers, OEMs, and customers. Since it is even hardly possible to estimate corresponding benefits as a result of prevented attacks, it is yet difficult to make a useful internal cost-benefit calculation.

Long Product Life Cycles. Since vehicular IT systems—in comparison to, for example, usual operating system software—have only limited possibilities for maintenance; simplicity, stability, and reliability of deployed hardware and software are obligatory requirements. Compatibility, portability, and reusability are further important requirements for all implemented security solutions, as it is not unlikely that a vehicular manufacturer runs out of certain microelectronics due to the typically rather short semiconductor product life cycles. Moreover, all security mechanisms and corresponding infrastructures have to be designed for proper operation during the complete life cycle of a typical vehicle (e.g., by flexible parameters, usage of long-term security values, foreseeing of upgrades), that means up to two decades.

Comprehensive Liability. As vehicular IT is often involved in highly safety-critical applications (e.g., driving assistances, crash prevention), they cannot released "without any warranty" and "at owner's risk" as most general-purpose computer software usually does. To provide operating safety and legal security, legally binding warranties are mandatory solely due to legal liability and government legislation. However, warranty statements can usually be given only based on complex and expensive internal and external certification procedures [Ame04]. Thus, corresponding documentation, models, tests, and assessments as well as the development process itself have to be prepared for possible certifications already at the beginning of every development process.

Interoperability and Compatibility. An important key factor for most IT security solutions is interoperability to existing (security) infrastructures and devices to enable end-users to integrate their existing devices (e.g., mobile navigation systems, smart phones, multi-

media players) as simple and holistic as possible. Closely related to interoperability is compatibility, which assures that updated or upgraded vehicular (security) functionality does not harm or lower other existing (security) functionality. However, due to the increasing complexity, functional dependencies, and mutual interactions (cf. Section 4.2.1), the verification of compatibility becomes even more complicated.

Usability and Limited Willingness for User Interaction. Any security measure becomes worthless if it is not (correctly) applied by the intended entities due to a bad or cumbersome usability. Regularly changing passwords for instance, would potentially increase the security of an authentication system, but often yield to quite the contrary. Since it is difficult for people to choose and remember a new password every month, they usually begin to choose much weaker "serial passwords" such as number series or month names. Moreover, many security mechanisms come with several inconveniences for the involved users (without a directly apparent benefit). However, the willingness of users to spend time and effort on a security mechanism that is difficult to understand, time-consuming or error-prone is very limited.

Various Patents and Regulations. IT security is subject to various patents, many different cryptographic laws, protection claims and regulations (e.g., import/export regulations [Koo07]), which can complicate the overall congeneric deployment of IT security measures required by the global automotive market.

5 Conclusion

IT security has become an accepted challenge for many vehicular IT applications. However, the realization of vehicular IT security measures can be seldom simply derived from existing (conventional) procedures. In fact, it mostly requires a very individual and well-adapted approach that carefully considers all the individual objectives and requirements of all the entities involved while efficiently incorporating all application-depending restrictions and specific boundary conditions.

This contribution has motivated the need for a proper security requirements engineering as an inherent part in the development of most vehicular IT applications. It has described how to identify the individual security objectives, how to derive the corresponding security requirements, and has indicated some helpful vantages and existing constraints characteristically in the automotive domain.

Based on a proper security requirements engineering, IT security can become the crucial enabler for many future vehicular applications while ensuring safety, reliability, and dependability for drivers, suppliers, and OEMs.

References

[AES+07] N. Asokan, Jan-Erik Ekberg, Ahmad-Reza Sadeghi, Christian Stüble, and Marko Wolf. Enabling Fairer Digital Rights Management with Trusted Computing. In *Proceedings of the 10th International Conference on Information Security, ISC 2007, Valparaiso, Chile, October 9 – 12, 2007*. Springer-Verlag, 2007.

[Ame04] Sandro Amendola. Improving Automotive Security by Evaluation – From Security Health Check to Common Criteria. White paper, Security Research & Consulting GmbH, 2004.

[And01] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[Ban03] The Bangkok Post. Computer Traps Thailand's Finance Minister Suchart. The Bangkok Post, May 13, 2003.

[Bro06] Manfred Broy. Challenges in Automotive Software Engineering. In *ICSE '06: Proceedings of the 28th International Conference on Software Engineering*, pages 33–42. ACM Press, 2006.

[CCD07] CCDB-2007-04-001. *Application of Attack Potential to Smartcards, Common Criteria Supporting Document and Mandatory Technical Document Version 2.3*. Netherlands National Communications Security Agency (NLNCSA), 2007.

[DoD85] DoD 5200.28-STD. *Trusted Computer System Evaluation Criteria (TCSEC)*. United States Department of Defense (DoD), 1985.

[Dri02] Kevin R. Driscoll. Safety in Automotive Industry. TTA-Group Forum, Munich, Germany, November 15, 2002.

[Eur07] The European Emergency Call Driving Group (eCall). `http://europa.eu.int/information_society/activities/esafety/forum/ecall/`, 2007.

[Fri04] Hans-Georg Frischkorn. Automotive Software – The Silent Revolution. In *Workshop on Future Generation Software Architectures in the Automotive Domain, San Diego, CA, USA, January 10 – 12*, 2004.

[ISO04a] ISO/IEC 13335:2004. *Management of Information and Communications Technology Security*. ISO/IEC, 2004.

[ISO04b] ISO/IEC 15446:2004. *Information Technology – Security Techniques – Guide for the Production of Protection Profiles and Security Targets*. ISO/IEC, 2004.

[ISO05a] ISO/IEC 15408:2005. *Information Technology – Security Techniques – Evaluation Criteria for IT Security*. ISO/IEC, 2005.

[ISO05b] ISO/IEC 17799:2005. *Information Technology – Code of Practice for Information Security Management*. ISO/IEC, 2005.

[ISO07] ISO/IEC 27002:2005. *Information Technology – Security Techniques – Code of Practice for Information Security Management*. ISO/IEC, 2007.

[Koo07] Bert-Jaap Koops. Survey of Cryptography Laws. `http://rechten.uvt.nl/koops/cryptolaw/`, 2007.

[MLO97] Terrence Mitchem, Raymond Lu, and Richard O'Brien. Using Kernel Hypervisors to Secure Applications. In *Proceedings of 13th Annual Computer Security Applications Conference (ACSAC)*, pages 175–181. IEEE Press, 1997.

[MN03] Jonathan D. Moffett and Bashar A. Nuseibeh. A Framework for Security Requirements Engineering. Technical Report YCS 368, University of York, UK, August 2003.

[MZ05] Norman Meyersohn and Tom Zeller. Can a Virus Hitch a Ride in Your Car? The New York Times, March 13, 2005.

[Pat99] Andy Patrizio. Why the DVD Hack Was a Cinch. *Wired News*, 2, 1999.

[PWW04] Christof Paar, André Weimerskirch, and Marko Wolf. Sicherheit in automobilen Bussystemen. In *Automotive – Safety & Security 2004, Stuttgart, Germany, October 6 – 7, 2004*. Gesellschaft für Informatik e.V. (GI), Shaker-Verlag, 2004.

[RG06] Klaus Rüdiger and Martin Gersch. In-Vehicle M-Commerce: Business Models for Navigation Systems and Location-based Services. In *Embedded Security in Cars: Securing Current and Future Automotive IT Applications*, pages 247–273. Springer-Verlag, 2006.

[RPH06] Maxim Raya, Panos Papadimitratos, and Jean-Pierre Hubaux. Securing Vehicular Communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, 13(5):8–15, 2006.

[Rus01] John Rushby. Security Requirements Specifications: How and What? In *Symposium on Requirements Engineering for Information Security (SREIS)*, 2001.

[Shi00] R. Shirley. RFC 2828: Internet Security Glossary. Technical report, GTE/BBN Technologies, `www.rfc-editor.org/rfc/rfc2828.txt`, May 2000.

[SJV+05] Reiner Sailer, Trent Jaeger, Enriquillo Valdez, Ramon Caceres, Ronald Perez, Stefan Berger, John Linwood Griffin, and Leendert van Doorn. Building a MAC-Based Security Architecture for the Xen Open-Source Hypervisor. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC)*, pages 276–285. IEEE Press, 2005.

[WFM+07] Peter Wilson, Alexandre Frey, Tom Mihm, Danny Kershaw, and Tiago Alves. Implementing Embedded Security on Dual-Virtual-CPU Systems. *IEEE Design & Test of Computers*, 24(6):582–591, 2007.

[WWW07] Marko Wolf, André Weimerskirch, and Thomas Wollinger. State of the Art: Embedding Security in Vehicles. *EURASIP Journal on Embedded Systems (EURASIP JES), Special Issue: Embedded Systems for Intelligent Vehicles*, 2007:Article ID 74706, 2007.