Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2007, Article ID 74706, 16 pages
doi:10.1155/2007/74706

Research Article

State of the Art: Embedding Security in Vehicles

Marko Wolf,[1] André Weimerskirch,[2] and Thomas Wollinger[2]

[1] *Horst-Görtz-Institute for IT Security, Ruhr-University Bochum, Universitätsstraße, 44780 Bochum, Germany*
[2] *escrypt-Embedded Security GmbH, Lise-Meitner-Allee 4, 44801 Bochum, Germany*

Received 19 October 2006; Accepted 13 April 2007

Recommended by Paolo Lombardi

For new automotive applications and services, information technology (IT) has gained central importance. IT-related costs in car manufacturing are already high and they will increase dramatically in the future. Yet whereas safety and reliability have become a relatively well-established field, the protection of vehicular IT systems against systematic manipulation or intrusion has only recently started to emerge. Nevertheless, IT security is already the base of some vehicular applications such as immobilizers or digital tachographs. To securely enable future automotive applications and business models, IT security will be one of the central technologies for the next generation of vehicles. After a state-of-the-art overview of IT security in vehicles, we give a short introduction into cryptographic terminology and functionality. This contribution will then identify the need for automotive IT security while presenting typical attacks, resulting security objectives, and characteristic constraints within the automotive area. We will introduce core security technologies and relevant security mechanisms followed by a detailed description of critical vehicular applications, business models, and components relying on IT security. We conclude our contribution with a detailed statement about challenges and opportunities for the automotive IT community for embedding IT security in vehicles.

1. INTRODUCTION

Information technology—we broadly define as being systems based on digital hardware and software—has gained central importance for many new automotive applications and services. The costs for software and electronics are estimated to approach the 50% margin in car manufacturing in 2015 [1]. Perhaps more importantly, there are estimates that already today more than 90% of all vehicle innovations are centered on IT software and hardware [2]. These applications are realized as embedded systems and range from simple control units to infotainment systems equipped with high-end processors whose computing power approaches that of current PCs. In premium cars, one can find up to 70 processors that are connected by several bus types and up to several hundred megabytes of embedded codes.

Not surprisingly, many classical IT and software technologies are already well established within the automotive industry, for instance hardware-software codesign, software engineering, software component reuse, and software safety. However, one aspect of modern IT systems has little attention in the context of automotive applications: IT security. Security is concerned with protection against malicious manipulation of IT systems [3, 4]. The difference between IT safety and IT security is depicted in Figure 1. Nevertheless, IT safety and IT security are interleaved fields, that is, some technical failure (safety issue) can be used to realize some malicious threat (security issue) and vice versa.

However, there are today niche applications in the automotive domain (e.g., immobilizers) that particularly rely on IT security technologies. Nevertheless, the majority of software and hardware systems in current cars is *not* protected against manipulations. The reason being that past car IT systems did not need security functions because there was only little incentive for malicious manipulation. Secondly, security tends to be an afterthought in any IT system, because achieving of the core function is often the main focus when designing a system. As can be seen for instance by the Internet development, implementing IT security afterwards, is normally doomed to failure.

The situation has changed dramatically, as we will state in this contribution with respect to the arguments given above. More and more vehicular systems need security functionality in order to protect the driver, the manufacturer, and the component supplier. Secure software update of electronic control units (ECUs), preventing chip tuning, preventing the unauthorized change of the mileage, or assembling nonoriginal parts are only some examples. Future cars will become even

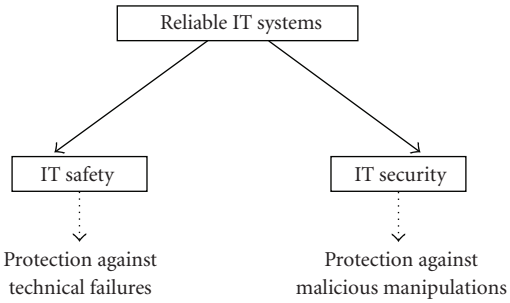

Figure 1: The relationship between IT safety and IT security.

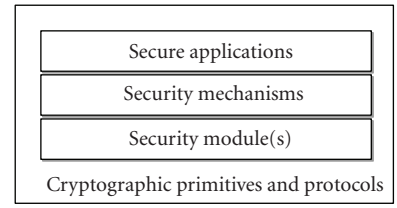

Figure 2: Layered security architecture to enable security-critical vehicular applications based on cryptographic primitives and protocols.

more dependent on IT security due to the following developments.

(i) An increasing number of ECUs will be reprogrammable and have to be protected.
(ii) Electronic antitheft measures will go beyond current immobilizers, for example, by protecting individual components.
(iii) An increasing number of legislative requirements (e.g., secure emergency call functions).
(iv) New business models (e.g., time-limited car functions or pay-per-use infotainment content) will be established.
(v) Vehicles will communicate with the environment in a wireless fashion that requires protected car-to-infrastructure communication.
(vi) Increasing networking of cars enables car-to-car communication that has to be protected against abuse and violation of privacy.

IT security will play an important role for several future automotive technologies and will even be an enabling technology for some future applications. The target platforms within cars that incorporate security functions are embedded systems, rather than classical PC-style computers. Some obvious differences in comparison to common PC-based environments are listed below.

(i) Embedded devices have small processors (often 8-bit or 16-bit microcontrollers) which are limited with respect to computational capabilities, memory, and power consumption. Hence, the usage of cryptographic primitives and protocols is limited.
(ii) Embedded devices mostly have only limited possibilities and limited bandwidth for external communication. Hence, the extent and frequency of external communication, for example, for internal updates, are limited.
(iii) Attackers of embedded systems have often physical access to the target device itself.
(iv) Embedded systems are often relatively cheap and cost-sensitive because they often involve high-volume products. Thus, adding complex and costly security solutions is not acceptable.
(v) It is costly to establish the necessary organizational aspects for security products, for example, one needs to adopt the production and life-cycle chain.

Hence, the technologies needed for securing vehicular applications mainly belong to the field of embedded security that differs from general IT security.

Outline

The topics discussed in this contribution give a state-of-the-art overview of IT security in vehicles. We start with an introduction of basic cryptographic functionality in Section 2, providing the theoretical framework for most security mechanisms in cars. We then point out the necessity of vehicular IT security while presenting attacks and attackers, deduce relevant security objectives, and indicate characteristic constraints within the automotive area in Section 3. As depicted in Figure 2, we subsequently introduce and explain each essential layer to enable security-critical applications within vehicles. Therefore, we first discuss the necessary security module in Section 4, followed by an overview of security mechanisms in Section 5 that are based on the security module. In Section 6, we present various current and future security-critical vehicular applications that rely on available vehicular IT security. We conclude our contribution with a detailed statement about challenges and opportunities for the automotive IT community for embedding IT security in vehicles.

IT security, however, comprises both technical and organizational measures. IT security systems always include security relevant organizational processes, and in many cases an IT security system is compromised due to organizational weaknesses. To enable secure automotive IT applications, complex and reliable organizational structures are required. Thus, organizational security has to be considered individually and additionally to all technical measures treated in our contribution.

2. CRYPTOGRAPHIC BACKGROUND

Besides security enhancing technologies such as filtering (e.g., firewalls), anomaly detection (e.g., intrusion detection systems), or vulnerability scanning (e.g., antivirus software), cryptographic primitives for data encryption and decryption, signature generation, and verification including the

necessary cryptographic protocols are the core of virtually all security-critical IT systems.

Understanding the basic functionality is essential for designing, analyzing, implementing, and assessing an IT security system. In this section, we therefore identify the basic security services that can be provided by cryptography followed by short introductions of symmetric- and public-key cryptographies, cryptographic hash functions, and cryptographic protocols relevant for vehicular security applications.

2.1. Security properties

Even though security depends on much more than cryptographic algorithms—a robust overall security design including secure protocols and organizational measures is needed as well—cryptographic primitives and schemes are in most cases the atomic building blocks of a security solution. In the following, we specify the security properties that properly combined cryptographic primitives and schemes are required to enable. Further reading can be found in [5, 6].

(i) *Confidentiality* (or *privacy*) is a service ensuring that information is kept secret from all but authorized parties.

(ii) *Integrity* is a service ensuring that unauthorized parties cannot modify system assets and transmitted information. Modification includes writing, changing, changing the status, deleting, and creating the transmitted messages. It is important to point out that integrity relates to active attacks as well as technical errors, and therefore it is concerned with detection rather than prevention. Moreover, integrity can be provided with or without recovery.

(iii) *Authentication* (more precisely *message origin authentication*) is a service concerned with assuring that the origin of a message is correctly identified. Note that origin authentication implies integrity; the opposite is not true.

(iv) *Identification* (more precisely *entity authentication*) is a service establishing the identity of an entity (e.g., a person, computer, credit card).

(v) *Nonrepudiation* is a service that prevents the sender of a message from denying commitments or actions.

(vi) *Access control* is a service restricting access to resources to privileged entities.

Security services can be achieved by employing the two most important cryptographic schemes: symmetric and asymmetric cryptographies. Symmetric cryptography provides the ability to securely exchange messages between two parties. This is especially important if the data should not be revealed to any third party. Authentication without nonrepudiation can also be achieved if the secret key is known only to the two parties. The second family of schemes, asymmetric or public-key algorithms, provides advanced functions such as digital signatures and key distribution over insecure channels. For common automotive applications, both symmetric- and public-key algorithms are used.

2.2. Symmetric-key cryptography

Symmetric-key cryptographic algorithms are the basic building blocks of any secure system that requires at least confidentiality. They are used to encrypt messages in bulk and to provide secure storage of data. In this kind of cryptographic algorithms, the keys used for encryption and decryption are the same for both communicating entities, and hence called a *symmetric cipher*. It can be considered as a locked box with the messages inside that is sent to the other party. If the other party has the right key to the lock, then the party can open and read all the messages in the box. The security of the symmetric cipher depends on the key (the algorithm is assumed to be public). The exchange of these keys between the parties should be done using a secure channel, for example, provided by a public-key cryptosystem.

Symmetric-key algorithms are mainly divided into two categories: *block ciphers* and *stream ciphers*. Block ciphers encrypt the messages in data blocks of fixed length, mostly 64 bits or 128 bits. Most well-known block ciphers are the data encryption standard (DES) [7], and the advanced encryption standard (AES) [8]. DES was the first commercially standardized block cipher with 64-bit data block size and 56-bit key size. The algorithm has been widely used because it was the only standardized and openly available algorithm extensively studied by the cryptanalytic community. There have been no major weaknesses found in the algorithm to date to practically break it other than the relatively small size of the key. This allows a brute force attack running through all the keys. DES finally expired as an US standard in 1999 and the National Institute of Standards (NIST) selected the Rijndael algorithm as the advanced encryption standard (AES) in October 2000. In the transition phase, triple-DES was approved as an FIPS standard [7]. The Rijndael algorithm [9] developed by Daemen and Rijmen was selected in an open challenge from a large set of algorithms submitted. AES [8] supports variable block and key sizes of 128 bits, 192 bits, and 256 bits to give a choice of different security levels based on its application. AES has been optimized for efficient software and hardware implementations.

Unlike block ciphers, stream ciphers encrypt a plain text bit by bit. The most famous example is the one-time pad (OTP) [10] encryption (also called Vernam cipher) which is the only known cipher which can be proven to be unbreakable [11]. The OTP works by bitwise XOR of the plain text with a one-time key, which is of the same length. The problem of having a secret key of the same length as the message to be transmitted over a secure channel makes OTP encryption inconvenient in practice. This shortcoming is overcome by using a pseudorandom generator as source for the secret key (but the unconditional security holds no more). Today's stream ciphers operate on a single bit of plain text (or a few bytes of data) being XORed to a pseudorandom key stream generated based on a master key and an initialization vector. Stream ciphers are especially useful in situations where transmission errors are highly probable because they do not have error propagation. They can be used when the data must be processed one symbol at a time because of lack

of device memory or limited buffering. Furthermore, stream ciphers mostly provide a higher throughput in comparison with block ciphers.

2.3. Public-key cryptography

The main function of symmetric algorithms is the encryption of information, often at high speeds. However, there are two problems with symmetric-key schemes.

(1) It requires secure transmission of a secret key, before being able to exchange messages.
(2) If in a network environment, each pair of users shares a different key, this will result in many keys.[1] Hence, this fact may result in problems handling the key management.
(3) After secure reception of a secret key, each party has to store its key securely for reuse.

The idea behind public-key (PK) cryptography can be visualized by making a slot into the locked box so that everyone can deposit a message (like a letter box). However, only the receiver can unlock the box and read the messages inside. This concept was first proposed by Diffie and Hellman [12] in 1976.

Public-key cryptography is based on the idea of separating the key used to encrypt a message from the one used to decrypt it. Anyone who wants to send a message to another party, for example, to Bob, can encrypt that message using Bob's *public key*. However, only Bob can decrypt the message using his *private key*. It is understood that the private key should be kept secret at all times, whereas the public key is publicly available to everyone. Furthermore, it is impossible for anyone, except Bob, to derive the private key from the public key (or at least to do so in a reasonable amount of time).

One can realize three basic mechanisms with public-key algorithms:

(i) key establishment and key exchange;
(ii) digital signatures;
(iii) data encryption.

In general, one can divide practical public-key algorithms into three major families.

(i) Algorithms based on the *integer factorization problem*: given a positive integer n, it is computationally hard to find its prime factorization, for example, RSA [13].
(ii) Algorithms based on the *discrete logarithm problem* (DLP): given α and β, it is computationally hard to find x such that $\beta = \alpha^x \bmod p$, for example, the Diffie-Hellman key exchange and the digital signature algorithm (DSA).
(iii) Algorithms based on *elliptic curves* rest upon the DLP on the algebraic structure of elliptic curves over finite fields. Elliptic curve cryptosystems [14, 15] are the most recent family of practical public-key algorithms, which have gained acceptance including standardization [16].

There are many other public-key schemes, such as NTRU or systems based on hidden field equations, which are not in widespread use. The scientific community is only at the very beginning of understanding the security of such algorithms. Despite the differences between their underlying mathematical problems, all three algorithm families have something in common: they all perform complex operations on very large numbers, typically 1024–4096 bits in length for the integer factorization and discrete logarithm systems, and 160–256 bits in length for elliptic curve systems (see also Table 1). This results in a poor throughput performance in comparison with symmetric ciphers. Nevertheless, public-key algorithms solve the key distribution problem in an elegant way, since the public part of the key can be distributed via an unsecured channel. Hence, one can establish a secure link between two parties without the need for an ulteriorly, previously exchanged secret. Thus, PK encryption is normally used for transmitting only small amount of data, like symmetric keys (see Section 2.6). Public-key algorithms are not only used for the exchange of secret keys, but also for the authentication by using digital signatures. Digital signatures are analogous to handwritten signatures. They enable communication parties to prove to a third party that one party has actually generated the message, also called nonrepudiation. The idea of the digital signature is appending a digital data block to the message that can be generated according to the message only by the person who signs it (like conventional signatures). Since the digital signature is a function of the message content and the private key, only the holder of the private key can sign the corresponding message. In practical terms, we use the private key for signing (thus only the holder of the nonpublic private key can sign a document) and the public key for the verification (thus everyone can verify the signature using the openly available public key). For practical implementations, using the RSA algorithm for digital signatures, a significant smaller public key[2] can be chosen to make the verification of an RSA signature a very fast and facile operation. Hence, RSA should be used in applications where the verification is done on the embedded platform and the signing on a personal computer or server. Instead, ECC should be used for applications where the embedded device performs encryption and signature generation as well as decryption and signature verification, since ECC is more efficient considering an application where the embedded device has to cover the complete public-key functionality.

2.4. Recommended key length

Table 1 puts the public-key and symmetric-key bit lengths in perspective. This recommendation assumes that in the near

[1] For a network with n users, $n \cdot (n-1)/2$ individual keys have to be shared afore.

[2] However, the private RSA key needs to have full length, for security reasons.

Table 1: Recommended key length for public-key and symmetric-key cryptographies.

Security	AES/DES	ECC	RSA
Short-term	64 bits	128 bits	700 bits
Middle-term	80 bits	160 bits	1024 bits
Long-term	128 bits	256 bits	4096 bits

future, there will be no unexpected (mathematical) attacks. PK systems need much longer keys, because of the attacks known today, which are more powerful than in the case of private-key primitives. However, choosing the appropriate key length depends much on the kind and security targets of the respective application. Highly security-critical vehicular applications such as digital tachographs, motor control units, or immobilizers have to provide *at least* middle-term security, whereas less security-critical applications such as personalized presets or customer information services could apply even short-term security. Although OEMs hardly provide any public information about applied security standards, we will provide at least two useful references providing key length recommendations for flash security [17] and for wireless car access [18].

2.5. Hash functions

In cryptography, hash functions are used in many applications, for example digital signatures, pseudorandom number generators, one-way functions, message authentication codes (MAC), and others. Hash functions compress a message of any length to a (nearly) unique string of fixed length, the so-called hash value or digital fingerprint. Hash functions are *one-way functions*, that is, for (almost) all given outputs y, it is impossible to find any input x such that $h(x) = y$. Hence, with a given input, a hash value can be computed, but it is computationally infeasible to compute the input if only the hash value is known. A *collision-free* function is a function where an attacker cannot find two inputs that compute the same hash value. Since hash functions map more than one value to the same hash, a collision cannot be prevented, but it has to be hard for the attacker to find a collision.

Nowadays, there are several families of hash functions. The MD family [19] and SHA family [20] are the ones mostly used. The MD family generates hash values up to 128 bits but suffer from serious flaws[3] making further use of the algorithm for security purposes questionable. The SHA family was developed by the NSA in 1995 (updated last in 2004) and generates hash values up to 512 bits. Attacks have been conducted also within the SHA family particularly for the widely used SHA-1 (160-bit hash value). No attacks have yet been reported on the higher SHA variants (256 bits and 512 bits), but since they are similar to SHA-1, researchers are worried, and are currently developing candidates for a new hashing standard.

[3] There exist algorithms to find a collision within minutes using a standard computer.

2.6. Cryptographic protocols

Two cryptographic protocols used for many automotive applications are *hybrid encryption* and the *challenge-response* protocol.

The major disadvantage of public-key primitives, when compared to symmetric-key schemes, is the arithmetic intensive operations that need to be performed. Hence, this can lead to a poor system performance. Even when properly implemented, all PK schemes proposed to date are several orders of magnitude slower than the most efficient symmetric-key schemes. Hence, in practice, cryptographic systems are applied as a mixture of symmetric-key and public-key cryptosystems in a hybrid fashion. Usually, a public-key algorithm is chosen for key establishment and then a symmetric-key algorithm is chosen to encrypt the communications, achieving in this way high throughput rates. As shown in Figure 3, the sender (Alice) first encrypts a symmetric-key K with the public key $\mathrm{PK}_{\mathrm{Bob}}$ of the receiver (Bob). Bob then decrypts K using his secret private key $\mathrm{SK}_{\mathrm{Bob}}$. Afterwards, both proceed their communication using a symmetric cipher with the previously shared K.

The challenge-response protocol provides entity authentication also called identification, that is, one communication party identifies itself to a second party. The identification can be provided by using knowledge, possession, or individual properties. The basic idea of the protocol is that one party challenges the second party, for example, by sending a random number. The challenged party then has to answer with the correct response. This correct response can be generated only if the second party has for instance some kind of knowledge, for example, the key for a cryptographic primitive. The party can use the key to encrypt the given random number and returns it to the challenger, thus proving the possession of knowledge without revealing it.

Figure 4 presents the challenge-and-response protocol using a symmetric-key algorithm. However, the protocol can also be implemented using public-key primitives. In Figure 4, Alice challenges Bob by sending a random number c. Bob encrypts c together with the identity of Alice and returns the response r. Bob is authenticated once the identity of Alice and the random number are correctly verified. Note that only Bob can response to the challenge correctly, since only Bob possesses the knowledge of the appropriate secret key K.

3. AUTOMOTIVE ATTACKS, SECURITY OBJECTIVES, AND CHARACTERISTIC CONSTRAINTS

In the following, we first provide an overview of specific attacks and attackers in the automotive environment that differ from common PC-based IT systems. We then deduce overall automotive security objectives along with the

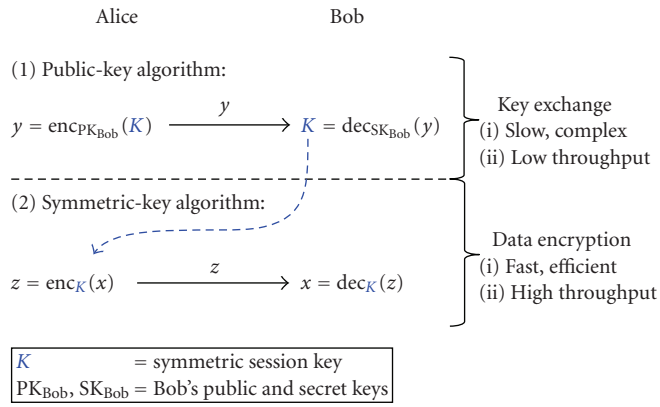

Figure 3: Hybrid encryption protocol.

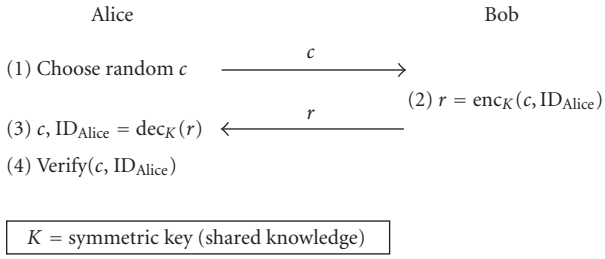

Figure 4: Challenge-response protocol.

characteristic automotive technical and organizational constraints.

3.1. Attackers in the automotive area

Today attackers within the automotive area usually either want to steal a vehicle or a certain valuable component (e.g., the navigation system) or—at owner's disposition—want to modify certain critical components. These modifications include for instance manipulation of the mileage for a higher resale value (reduced mileage) or a higher tax return (increased mileage), manipulation of the motor control unit (chip tuning) for unauthorized driving parameters, or manipulations of the tachograph to circumvent legal driving restrictions or to conceal potential previous infringements. With future electronic applications (cf. Section 6) such as electronic license plates, event data recorders, car communication, and copyrighted infotainment, misuse potentialities will obviously increase further. Finally, there exist nonnegligible, partially quite extensive, efforts to steal competitors' expertise and intellectual property in order to advance own developments or, more likely, to illegally produce marketable counterfeits.

Since automotive IT systems, in comparison to common (PC-based) IT systems, imply specific characteristics, attacks on vehicular IT systems differ from attacks on usual computer systems. Attackers of a computer system seldom have physical access to the target system, whereas attackers in the automotive sector mostly have physical access to all built-in electronics. If there are no further protection measures integrated, attackers can manipulate or replace all built-in components. Moreover, afterwards discovered vulnerabilities are much harder to fix once hundreds and thousands of vehicles are sold already. Finally, automotive attacks are usually "offline attacks," where attackers have almost unlimited time and unlimited trials to succeed.

According to the two different attacking objectives—theft and modification—we identify four different groups of attackers as depicted in Table 2. In case of theft, the thief may have considerable technical expertise and some appropriate tools. However, a thief usually has only limited physical access and limited time. Within the attacking group having systematic modifications in mind, three typical kinds of attackers can be identified. The first group includes individuals such as the car owner. They normally have only little technical expertise, a few appropriate devices, as well as restricted financial resources for applying an attack. Skilled (OEM) garage employees are the second group of attackers. They have appropriate tools, have the necessary technical expertise, and are mostly endued with insider information. They even would invest money if an attack promises appropriate revenues, that is, if the attack can be scaled to many automobiles easily. The third group of attackers includes concurrent manufacturers, counterfeiters, and organized crime that may have immense technical and financial resources limited only be the potential economic gain. The motivation of this group is to gain competitors intellectual property (IP) or to exploit the outcome of an attack commercially, for example, by selling counterfeits or providing tools and expertise in the Internet.

Since the group of counterfeiters and organized crime is the most powerful and dangerous one, all actions to protect automotive IT should try to resist in particular attacks from these in such a way that the costs of a successful attack will exceed the potential gain. More concrete, a single successful attack on an automotive device must not scale to break also all other devices, for example, by revealing a global identical secret.

Table 2: Attackers in the automotive area.

Target	Systematic modification			Theft
Attacker	Individual, owner	Mechanic, garage personnel	Organized crime, competitor, faker	Thief
Technical resources	Varied (Generally low)	High	Very high	Varied
Financial resources	Low	Medium	Very high	Low
Physical access	Full	Full	Full	Limited
Risk	Low	Medium	Very high	Medium

3.2. Automotive attacks

This section provides an overview of specific hardware and software attacks in the automotive environment that typically differ from attacks on common PC-based IT systems.

3.2.1. Attacks on automotive hardware

Attacks on automotive hardware comprise attacks to replace critical components with unauthorized components or to illegally modify existing components. Usually, most hardware components provide, beyond some more or less sophisticated tags, no further protection mechanisms. They can be easily cloned, modified, or replaced by unauthorized components. However, a few critical components such as the tachograph, the speedometer, or airbags provide some basic (cryptographic) mechanisms to prevent or at least to detect unauthorized modifications, replacements, or misuse.[4] In such cases, hardware attacks aim at the circumvention or breaking of these protections by readout of secret keys, deactivation of alarm channels, or wiretapping their operation or communication.

3.2.2. Attacks on automotive software

Today's vehicles hold several dozen electronic control units (ECU) that control almost anything such as air conditioning, electric windows, engine, and break system. Several of these ECUs allow downloading updated program and data code to apply bug fixes, to improve existing functionality, to renew underlying data, or to install/activate new software features. The software update might be performed over a diagnosis channel, other available communication channels such as Bluetooth and GSM, or by using a storage medium such as a CD-ROM or a USB device.

However, present automotive IT systems are mostly unprotected against malicious software attacks. Often, for example, ECUs memory can be accessed without any further restrictions using their regular interface. Other may be compromised employing unprotected diagnosis or communication interfaces. At last, all ECUs without further measures for tamper resistance can be dismounted and analyzed offline using sophisticated analysis equipment. Obfuscation techniques[5] and pure software encryption (without hardware support) provide only minimal additional protection, since all programs have to be decrypted during runtime, and hence will be stored and decrypted at one point. The program code can then be read out and analyzed by attackers with only moderate technical understanding. Moreover, encryption keys are mostly stored somewhere unprotected or can be guessed easily. Disabling even sophisticated software protection measures by reengineering the "decisive validation branch" within the binary enables circumvention of almost all available software protection mechanisms [22].

Important (software) security vulnerabilities could also originate from inadequate OEM-internal software protection management. Thus, employees should not be able to reveal software to competitors or other unauthorized persons (unconsciously or maliciously) if adequate organizational security precautions are established and executed.

3.3. Overall security objectives

To guarantee road safety and operational reliability of vehicles and to sufficiently protect business models based on the security of the vehicular platform, we define the following overall automotive security objectives.

(i) *Confidentiality of data*: unauthorized access to protected data must be infeasible.

(ii) *Integrity of data*: unauthorized modification of data must be infeasible or at least detectable.

(iii) *Hardware and software integrity*: unauthorized modifications to vehicular hardware and software must be infeasible or at least detectable (by the vehicle).

(iv) *Availability*: authorized hardware and software components must have proper access to their dedicated data and services.

(v) *Uniqueness*: unauthorized cloning of a hardware components must be infeasible or at least detectable as nonauthentic.

3.4. Technical constraints

The application of complex IT systems in automotive environments is subject to some characteristic technical constraints.

[4] Night vision devices for instance, already available for premium class vehicles, are mandatory [21] protected against unauthorized, nonautomotive usage, for example, as military equipment, by terrorists or guerilla forces.

[5] Still used to "protect" for instance mileage information.

Automotive computing resources are—in comparison to usual computer systems—rather limited. Nevertheless, automotive applications are often required to provide (hard) real-time capabilities. This leads to severe requirements on complexity, memory size, and runtime efficiency for automotive implementations that moreover often have to cope with lots of specific architectural restrictions.

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.

Moreover, vehicular IT systems usually have only limited communication resources to, for example, exchange cryptographic keys or updating software. Thus, virtually all vehicular functionalities have to work properly even with an external communication functionality severely limited in capacity and frequency.

Since typical computer users can mostly employ ergonomic input and output devices, users within the automotive environment are restricted to only little ergonomically designed peripheral devices. To demand only a minimum of user interactions, virtually all vehicular applications are required to run almost completely autonomously.

3.5. Nontechnical constraints

Beyond the technical constraints, automotive IT systems are also subject to some particular organizational and legal constraints that may substantially differ from legal constraints for usual computer systems.

A possible public key and certificates infrastructure (PKI) for instance requires complex and costly organizational structures, particularly within the automotive context with a multitude of involved parties (e.g., manufacturer, supplier, OEM, garage personnel, content provider, etc.) and only limited (end-user) security understanding.

Another important key factor is interoperability to existing infrastructures and devices to enable end-users to integrate their existing devices (e.g., mobile navigation systems, smart phones, multimedia players, etc.) as simple and holistic as possible.

Since vehicular IT systems—in comparison to, for example, usual operating system software—have only limited possibilities for maintenance, compatibility, stability, safety, and reliability of deployed hardware and software are obligatory requirements. In particular, the corresponding support infrastructure must be available during the complete typical life cycle of the vehicle, that is, up to two decades.

Finally, as vehicular IT systems are often involved in highly safety-critical modules (e.g., steering lock, drive-by-wire systems), they cannot be released "without any warranty" and "exclusion of any damages" as most PC software usually does. For providing operating safety and legal security, legally binding warranties are mandatory. However, warranty statements can usually only be given based on complex and expensive internal and external certification procedures [23]. 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 any development process.

4. SECURITY MODULE

security module, which is also called a security anchor, provides necessary security relevant methods such as encryption and decryption, generation and verification of signatures, hashing, and secure storage of cryptographic keys. Such a module might be implemented in software or hardware. Clearly, a hardware solution provides higher performance and a far higher security level.[6] It is possible to deploy a single central security module in a vehicle (e.g., at a central control unit) or to implement it in each control unit that has a need for security. In the first case, a hardware implementation is appropriate to securely protect numerous critical assets; whereas in the latter case sometimes a software implementation could be adequate.

A security module must fulfil the following requirements.

(i) *Unclonable*: a security module must be unclonable. It is desirable to bind the identity of a vehicle to the security module in such a way that it can neither be faked, or manipulated, nor cloned. In addition, it must be impossible to install the security module in another car in order to change its identity.

(ii) *Secure key storage*: a security module must be able to store keys in a secret and protected way. It must protect secret keys from being read and public keys from being altered.

(iii) *Secure computations*: the security module must be able to securely (and efficiently) perform cryptographic operations to prevent leakage of cryptographic secrets into unprotected areas.

(iv) *Alarm channel*: in case of a security breach, the security module must be able to give notice. For instance, such an alarm channel might be provided at diagnosis.

A security module can be based on a customized security controller, a trusted platform module (TPM), or an FPGA. A TPM provides a compatible standard interface more suited to the PC office world, whereas a customized security controller approach can be adapted in a flexible way. Both approaches provide a highly secure computing environment as well as secure key storage. An approach based on FPGAs provides a very flexible way at higher cost. Table 3 summarizes the assets and drawbacks of different hardware solutions.

Using a security module purely based on software, runtime attacks exploiting available software interfaces can be usually avoided, if an implementation as small as possible

[6] Note that pure software security mechanisms can often be broken very easily [22], and thus provide only little protection of the corresponding control unit.

Table 3: Hardware security module.

	Trusted platform module (TPM)	Customized security controller	Field-programmable gate array (FPGA)
Standardized	Yes	No	No
Flexibility	Very limited	Yes, until release	Yes, even after release
Cost	Medium	Low (high volumes)	High
Security level	High	Adaptable	Medium-high

and secure[7] is used. Runtime or online attacks are limited to use given software interfaces and, for instance, try to inject malicious code. However, hardware modifications based on manipulation, exchange, and addition of hardware components probing communication lines cannot be prevented (or even detected) by pure software security modules. Applying solutions based on a hardware security module and plausibility checks, most attacks can be at least detected.[8] Hence, the main achievements of a security module are as follows.

(i) A single security module might save code size, and hence even cost.

(ii) A solution based on a software security module is able to prevent at least runtime software attacks (such as injecting malicious code).

(iii) A solution based on a hardware security module is able to prevent software attacks and detect hardware-based attacks (such as hardware manipulation).

5. SECURITY MECHANISMS

In the following, we present mechanisms based on cryptographic methods and the security module that enable securing components and business models described in the subsequent section. We start by presenting mechanisms to ensure hardware and software integrities as well as to secure communication channels.

5.1. Hardware protection

A facile way of providing a basic protection against hardware manipulations can be achieved by mechanical countermeasures deploying special component constructions. Such special constructions could be proprietary constructions that fit only into cars of a single manufacturer or constructions that require proprietary (not publicly available) tools and equipment. However, that solution is uncomfortably and provides only minimal hardware security.

More reliable approaches [24] for detection of faked or bogus vehicle components use small computing tags attached to each crucial component in order to logically bind security and safety related parts to a specially protected central security module. Such component identification schemes rely on the tamper-evidence of the computing tags that are tightly (nonremovable) integrated into critical components that can communicate with each other and on the tamper resistance of the central security module. The component identification protocol works even without the need of a central tamper resistance security module by distributing its task to the, of course more powerful, computing tags.

To protect hardware (and particularly hardware IP) effectively, all critical hardware cores have to be integrated completely into a single protected chip. Although there are (physical and chemical) methods to comprise even such a system on a chip (SoC), these are highly sophisticated and expensive methods. Thus, today attacks on SoC hardware can comprise only small amount of data and are not applicable to large amount of hardware. However, if the outcome is worthwhile enough, for example, if an SoC contains a globally similar secret key that easily enables fraudulent manipulation on a large scale, even sophisticated and expensive attacks are feasible.

5.2. Software protection

In order to provide effective software protection,

(1) only original software must be accepted by the vehicle: no manipulated or malicious software must be downloaded to the car. In particular, no software must be successfully downloaded to the ECU that alters the defined behavior of the vehicle (e.g., due to software version conflicts);

(2) only authenticated parties are able to alter data, for example, parameters, stored in the vehicle.

Furthermore, the following is desired for an actual security design:

(i) the compromise of a single control unit does not affect the entire system, that is, a successful attack does not scale;

(ii) the required computational performance on the side of the control unit will be minimal.

A solution for this problem in general is quite simple. Based on digital signatures, the issuer of the software signs the program code and the control unit in the vehicle verifies

[7] The ideal case would be a software security module that could be verified formally.

[8] Assuming sufficient time and money, even hardware attacks cannot be prevented at reasonable cost in a vehicle. Thus, a single successful attack on a security module must not scale to also break all other security modules.

Table 4: RSA signature verification on ARM7TDMI at 40 MHz.

	Code size	Time
SHA-1 hashing	1132	680 kB/s
RSA exponentiation w/small public key	2368	11 ms
RSA verification (16 kB code)	3500	34 ms

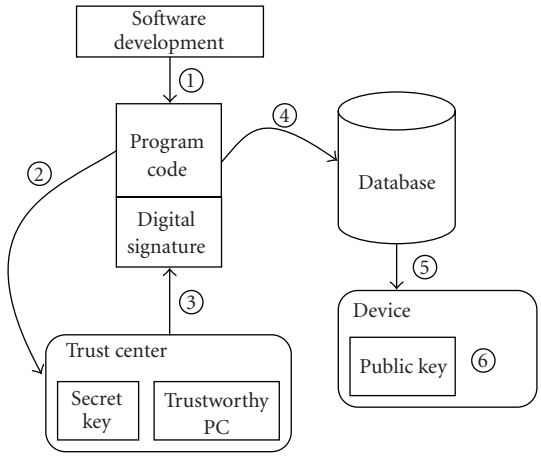

Figure 5: Secure software download.

it. Hence, the issuer holds a secret key for signing the program code and the control units hold the corresponding public key for verifying it. This is depictured in Figure 5 in more detail. First, the software is developed. Once it is finished (Step 1), the program object code is passed to a trust center (Step 2) that signs the object code using its secret key. The signature is then passed back and attached to the program object code (Step 3). The package of code and signature is now stored in a database (Step 4) that might hold versions for different control units. Finally, the appropriate program code is downloaded to a control unit (Step 5) and verified by means of the public key stored in the ECU (Step 6).

One can see that security objective (1) is clearly fulfilled: Only a legitimate authority can issue an appropriate signature for program code that will be accepted by the vehicle, that is, only authentic software will be accepted by the vehicle. Most of today's cars already provide a mechanism for downloading software. Hence, only a mechanism for verifying the signature is additionally needed in the vehicle. For signature verification, RSA is an appropriate fit since it allows very fast signature verification. This can be implemented in software. Some performance values of our implementation are displayed in Table 4. Note that for the signature verification, first the program code needs to be hashed and then an RSA exponentiation is performed. The last column displays the overall time for the signature verification at the example of 16 kB of program code, which is a typical size for a vehicle control unit.

On the server side, the key management and the organizational security must be thoroughly organized. Latter aspect includes organization of who has access to sign program code and how the process of signing is performed and recorded. However, there is no full-sized public-key infrastructure (PKI) necessary. It is sufficient to issue a single private/public key pair such that the private key is stored in the trust center and the public key in the control unit. The trust center might simply be a PC that is disconnected from any computer network and a secure smart card that holds the secret key. If a finer-grained approach is desired, a key pair for each control unit type, for each production year, or each production location might be applied. No certificates that induce overhead are required, though. The ECU only needs to store a public key such that no secret information is stored here. However, this public key must be protected from manipulation. Otherwise, an adversary could replace this key in the ECU and then induce any manipulated software.

Security objective (2) can be fulfilled by a simple challenge-and-response mechanism as presented in Section 2.6. The vehicle and an external party (e.g., a standard PC) share a secret key. The parties then run a challenge-response scheme in order to prove that the external party knows the secret key. After a successful run, the external party can access the vehicle's data. However, it is crucial that a well-defined interface is specified. For instance, it is reasonable to protocol all changes in a log file and give access only to nonsafety critical data. Using a symmetric-key management is reasonable—each ECU knows an individual symmetric secret key shared with the third party. This third party might be the car manufacturer storing all keys in a protected database.

Protecting the key of the ECU is crucial. If an adversary is able to read out or replace the key, he might be able to manipulate program or data code. Thus, virtual software protection can be achieved only by applying hardware-assisted approaches employing a security module as described in Section 4.

Nevertheless, there also exist mechanisms that try to complicate utilization or at least try to help identifying the origin in case a software could be successfully read out by an attacker. In order to make decompiling and reengineering of program binaries more difficult, programs known as "obfuscators" convert source code, object code, or both, into obfuscated code, making the result overcomplicated, and thus far less readable and almost impossible to understand by a human being. However, obfuscation [19, 25] only increases the difficulty for reverse engineering, limits portability, and is regarded as "security through obscurity." Digital watermarking [26] or fingerprinting are techniques which embed (visible or invisible) information into a digital content (software or data) that cannot or only hardly can be removed or modified. Original owners then can use tools to extract the embedded information to detect, for example, the origin of an illegitimate copy or tampering. However, these already exist technologies to abolish respective restrictions for both mechanisms. Thus, such mechanisms cannot replace a proper hardware-based software protection.

5.3. Secure communication

Until now, vehicles did communicate to the outside world only rarely. The communication channels were mainly provided for diagnosis purposes by proprietary methods. Now mobile devices, in particular cell phones, are connected to the vehicle's systems by a cradle and recently also by a wireless Bluetooth channel. The software download described above can also be seen as a communication channel. The vehicle manufacturers start to open more and more communication channels to the car such that vehicles are about to become more open. Hence, sophisticated strategies for secure communication are necessary. There are different general communication facets to consider here.

(i) *In-vehicle communication*: communication inside of a vehicle, for example, to link a mobile device to the vehicle's head unit or to allow communication between head unit and anti-lock braking system (ABS).

(ii) *Vehicle-to-vehicle communication*: communication between vehicles, for example, to exchange data about road conditions or traffic jams.

(iii) *Vehicle-to-infrastructure communication*: communication between vehicles and the infrastructure. For instance, sensors embedded in the road report icy streets and traffic lights forward their light phase.

The vehicle-to-vehicle as well as vehicle-to-infrastructure communications make possible a variety of new fascinating scenarios. For instance, cars communicate to each other in order to transmit information about imminent dangers (traffic jams, accidents, or sudden weather changes), traffic controls, or free parking spaces. In addition, it would be possible to group together cars on a highway and make driving more comfortable and safe. Digital contents such as multimedia files could be downloaded to a car when connected to the Internet (e.g., at a gas station) and navigation data could be updated at the same time. The license plate can be replaced by an electronic license plate that transmits the license number such that it can be used for various purposes, for example, wireless payments or automatic gateway access control.

There are various requirements to implement such scenarios. These comprise both technical and organizational aspects as follows.

(i) *Message integrity*: altered messages must be detected by the vehicle.

(ii) *Message authenticity*: origin of a message from a valid source must be verifiable by the vehicle.

(iii) *Privacy*: vehicles must not endanger privacy of the driver and owner. For instance, a GPS receiver or an electronic plate must not be used to track a car. On the other side, it is often desirable to allow authorities to access this data in a well-defined way.

(iv) *Efficiency*: if cryptographic algorithms are involved, these must often run extremely fast to allow real-time behavior. For instance, a warning message about an imminent danger must be processed immediately.

Secure communication is mainly based on encryption and authentication in order to provide confidentiality and authenticity of exchanged data. While authentication is necessary in most scenarios in order to make sure that there are no malicious messages induced to the network, confidentiality might often be less important. For instance, a warning message should be authentic but not confidential. Clearly, a vehicle should implement a balanced security policy in such a way that it reacts in a reasonable way based on external input, internal input (such as internal sensors), and most important the drivers decision. Based on the external infrastructure, protocols that are more sophisticated can be implemented. For instance, location-based protocols [27] as well as time-based protocols can be implemented.

There is a variety of literature available about secure communication. See [28] for in-vehicle communication and [27] for car-to-car and for car-to infrastructure communications. Furthermore, there are a variety of standardizations and on going projects dealing with such topic such as Car-2-Car Communication Consortium [29], Network on Wheels (NOW) [30], CIVS [31], SAFESOPT [32], and IEEE 1609.2 [18].

6. SECURITY-CRITICAL VEHICULAR APPLICATIONS

Several vehicular applications provide security features or are security relevant. These applications are usually based on a security module and security mechanisms as described before. In order to properly realize such a security-critical vehicular application the following steps have to be done:

(1) analysis of valuable attacking targets and all relevant attackers to create the corresponding attacker model (cf. Section 3.2);

(2) establishing the corresponding security objectives (cf. Section 3.3);

(3) design of a proper security module(s) capable to successfully fend off the attacker model and fulfill the security objectives derived before (cf. Section 4);

(4) implementation of the required security mechanisms, based on the afore-designed security module(s), the security-critical application builds on (cf. Section 5).

We would like to point out that once a security module and the corresponding security mechanisms have been implemented properly, this base can be easily reused to protect also other future security-critical applications[9] with almost no additional cost.

In the following, we give an overview about current and future security-critical vehicular applications that can be protected in this way.

[9] Provided that the added application has the same attacker model.

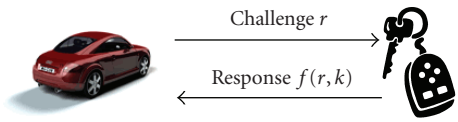

Figure 6: Electronic immobilizer.

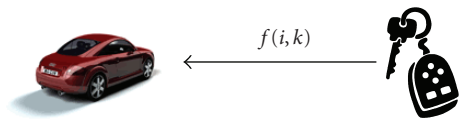

Figure 7: Remote key.

6.1. Electronic immobilizer

The electronic immobilizers as well as the keyless entry to a vehicle are probably the oldest applications of cryptography in vehicles. The electronic immobilizer usually works in the following way. The vehicle sends a challenge to a passive batteryless transponder integrated in the vehicle key, which then answers by a response. Transponder and vehicle share a secret key. Only if the transponder knows the secret key, then the vehicle will start. Hence, a vehicle's key that has the appropriate physical properties (i.e., that is an exact physical copy of the original key) but does not know the secret cryptographic key will not make the vehicle starting. This is depictured in Figure 6. Here, f is a cryptographic function such as a keyed hash function that takes as input the challenge r as well as a key k and returns the response. A general approach for an electronic immobilizer was presented in [33].

For the electronic immobilizer, attacks at the hardware layer must be considered. Such a hardware attack can never be prevented at reasonable cost. The goal is to make such an attack impossible for a rational attacker, that is, the cost of an attack will exceed the gain of the stolen car on the black market. Hence, the goal is to achieve an economic security. A hardware security module is an appropriate platform to provide such security goals. It is able to securely store the secret keys and to bind the key's transponder to the vehicle by means of the security module. The immobilizer binds a crucial control unit (usually the engine control unit) to the vehicle's key. Hence, the engine control unit is only activated if the proper key is presented. There are several weaknesses though. The crucial control unit can simply be replaced by one that is always activated, that is, by one that implements exactly the same functions but without the key verification. Hence, avoiding malicious software updates of the firmware is absolutely necessary. Furthermore, a so-called Mafia attack is possible. Here, the vehicle's signal is forwarded over an external channel (say, a wireless LAN) to the vehicle's key. This is in particular dangerous in combination with a keyless entry system where an adversary establishes a channel between the victim's vehicle and the victim such that the vehicle's doors unlock and the engine starts. Usually, once the vehicle starts, the engine will not turn off even if the key's signal is lost.

The later attack can hardly be avoided on a cryptographic layer. A countermeasure is to use the so-called distance bounding (see, e.g., [34]). Here, the protocol can make sure that the vehicle's key is inside of a well-defined geographical area. However, due to timing problems and wavelength of the transponder, this might be too imprecise. Further countermeasures can be provided on the physical level. For instance, multifrequency hopping is already applied today for such. Clearly, each electronic immobilizer can be compromised. However, the objective is not to set up a perfectly secure system, but an economical secure system—breaking a single vehicle should be more expensive than the gain of the attacker.

6.2. Keyless entry

The remote key entry works in a similar way as the electronic immobilizer. Here the key is equipped with an active battery-powered transponder. When pushing the button of the vehicle's remote control key, the vehicle will unlock the doors. Therefore, the so-called rolling codes are used. Both vehicle and key share a secret. Each time the key's button is pushed, a new secret is derived by the key and sent to the vehicle. The vehicle can compute the same transition and compare the two values. If they are equal, the vehicle unlocks its doors. Certainly, the key's button might be pushed several times. The vehicle then repeatedly computes the internal value and compares it. It repeats this, say, a hundred times before giving up. This is shown in Figure 7. The remote control as well as the car hold a counter i that is increased by one after each application.

Modern cars are equipped with a keyless entry system. Here, the driver carries a key-card in his pocket. Once he approaches the vehicle, the doors are unlocked. This system usually works as follows. The protocol starts when the door handle is touched. The vehicle then transmits a signal. Once the key card approaches the car, it will be detected by the vehicle and then responds to the car. Then a cryptographic challenge-response method is carried out. Note that the key card usually is comparable to a passive or battery-powered radio frequency identification (RFID) tag.

Attacks on the remote key entry and keyless entry system are located at the protocol and physical transmission layers. An attacker might try to compromise the secret key or to replay a message in such a way that it unlocks the doors. However, it can be assumed that there is no physical breach—an adversary could otherwise just smash a window. Hence, such systems can be designed in a secure way using traditional cryptographic schemes. However, attacks on the physical transmission layer such as the Mafia attack have to be considered carefully since they are inherent for any such scheme and can hardly be prevented by cryptographic means.

6.3. Digital tachograph and event data recorder

Digital tachographs and the so-called event data recorders are a well-considered security relevant component in

vehicles. Manipulation of tachographs has a serious safety and economic impact. Today, it is assumed that about a third of all used cars was manipulated regarding the tachograph counter, for example, in order to achieve higher prizes on the used markets. In the case of trucks, the attacker usually is the truck driver or the owner, who tries to circumvent rest periods, speed limits, and law regulations. However, recently several law regulations were introduced to stop such misbehavior. For truck tachographs, there was a European law introduced [35] concerning the required security level according to the ITSEC security standard and specification of the involved processes. This leads to several security certified truck tachographs. Furthermore, for vehicles in Germany, a law was introduced making any change of the tachograph counter liable to penalty.

The attacker of this system is usually the owner (company) or the driver of the vehicle. Hence, he has full physical access to any component and unlimited time for an attack. An attack on the hardware level is usually performed and the goal is to achieve an economic gain. For standard personal vehicles, it is almost impossible to prevent a manipulation at reasonable cost, whereas for high-cost trucks it is possible up to a certain point. In both cases, the first objective is to detect a manipulation though. Hence, some kind of security module is required. This might even involve a security controller such as a smart card controller as described by the European law regulation for truck tachographs.

Each truck is equipped with a motion sensor that receives input of the gearbox and transmits signals to the tachograph on an encrypted channel. The main objective is to record the truck drivers' behavior and working hours. The European law enforces that every truck driver uses a personalized smart card when driving the truck by inserting it to the digital tachograph. Clearly, privacy is a crucial aspect here. Hence, there are four kinds of smart cards provided. Besides the truck driver's card, there is a card for each company that owns trucks, for workshops that maintain the trucks, as well as for police authority. The smart cards are issued with keys that are organized by European wide key management hierarchy.

The security of the system can only be provided by a combination of technical and organizational means. For instance, it is possible to manipulate the gearbox such that the motion sensor receives false input. Therefore, integrity of the motion sensor and the gearbox must regularly be verified by police authorities. Once again, attacks that manipulate hardware components cannot be avoided by technical means, but they can be detected by a combination of technical and organizational means.

6.4. Counterfeit and expertise protection

Today, large amounts of OEM's capital investments are spent on software and electronic development [1] that—without further protection—can be simply copied, analyzed, and reused by simply buying the corresponding components or vehicles. Thus, reliable counterfeit and intellectual property (IP) protection should prevent copyright infringement or expertise theft by potential competitors and particularly to prevent mass production of unauthorized counterfeits of vehicle components.

(i) *Counterfeit protection*: illegal produced replacement parts cause a worldwide loss of about 3 billion dollars [36] per month. The professional organized manipulation of automotive electronics [37] causes considerable damage to the manufacturers and to the economics by unwarranted claims, brand damage, and undermined business models. Moreover, counterfeits endanger the safety of all motorists and cyclists. Traditional methods to prevent counterfeits use tags, for example, holographic stickers that are supposed to be unforgeable. However, there exist illegal businesses that create boxes, labels, and other significant trademark logos and emblems to let counterfeits look like real parts [38].

(ii) *IP and Expertise protection*: automotive OEMs and suppliers always have a comprehensible interest to find out valuable expertise from their potential competitors. Moreover, even though intellectual property rights are legally effective in most countries in the world, there exist large domestic markets, such as China, where IP thefts and infringements are virtually nontriable. Therefore, expertise leakage and IP theft are a serious problem. Today mostly for software and firmware, but even complex hardware, can be copied when it is profitable enough. Expertise leakage and IP theft have to be tackled primarily applying organizational security measures such as scrutinizing potential partners and preventing employees from unintentional (or intentional) exposures. However, there exist also (cryptographic) technologies that can help protecting IP and expertise or making a theft or leakage at least detectable.

6.5. After-sale business applications

Embedding security in vehicles enables various new and interesting business models previously not possible. Particularly, it enables business models where all involved parties (OEMs, suppliers, and customers) can benefit from. In the following, we present three exemplary business models made possible by progress in vehicular security.

6.5.1. Feature activation

The production of vehicular components moves from various small charges of different individually adjusted components towards large-scale production of only a small number of uniform standard components. Thus, today many of the various vehicle versions internally consist mostly of the same components. On the other hand, providing manifold individual vehicle configurations is crucial even now. To solve

this opposing requirements, car manufacturer could build parts identical in construction cost-efficiently with most features already built-in, but individually activated. Moreover, it is possible to individually activate (or deactivate) built-in hardware components or software after sales for an additional charge that would furthermore bind the customer long term to the OEM. Features that would be capable for after-market activation could be, for instance, special setups for engine, gear or chassis control, enhanced board computer and comfort diagnosis functions, additional driving assistance and infotainment capabilities, or certain personalization and individualization features. However, capable security measures are required to prevent unauthorized feature activation that may undermine the underlying business model.

6.5.2. Infotainment

Maybe the most exciting new applications in the automotive are driven by new infotainment business models distributing digital content. The area ranges from individual software upgrade packages, OEM premium content, newscasts up to various multimedia files including music, video, or games. Today, already most medium-sized cars are equipped with multimedia capable on-board computers and radio systems. Upcoming integrated wireless broadband communication promises a brisk market for automotive-related on-demand sales. Embedding a reliable digital rights management (DRM) enables business models for usage-metered and on-demand utilization of digital contents, software, and even hardware beyond the classical lump-sum model. Some possible examples are provided below.

(i) *Time-limited utilization*: up-to-date navigation and traffic data may be available on demand for any place in the world (e.g., only for a two-week vacation trip in the respective area).

(ii) *Quantity-limited utilization*: movies, music tracks, or games can be bought for an n-times repeated utilization.

(iii) *Device-bound utilization*: extra software can be installed on a particular device or a particular vehicle only. Certain car functions are performed only via a certain authentication device such as a driver's key, dealer token, or personal cellular.

(iv) *Usage-metered utilization*: navigation routes can be charged for their actually used length. Movies or music tracks can be charged for the actual viewing time.

(v) *Subscription services*: audio, video, or information broadcast services can be received as long as a valid subscription to the corresponding service exists.

Furthermore, almost arbitrary combinations are possible. For instance, an afterwards activated enhanced comfort sensor (e.g., tire air pressure sensor) may be enabled as free sample for 4 weeks. Business models using digital content that has usage or access restrictions are only possible with a secure and reliably implemented DRM system. As it could be seen in various (nonautomotive) DRM scenarios such as pay-TV, online music stores, or video game consoles, having no such secure module, the business model will certainly fail.

6.6. Location-based services

Offering services based on the vehicle's current location provided by a built-in GPS or GSM receiver together with a wireless communication device enables various safety, management, and business applications.

(i) *Automatic emergency call (eCall)*: the first popular location-based service would probably be the automatic emergency call, mandatory for all new vehicles within the European Union from 2009. As proposed by the eCall Driving Group [39], in case of emergency the eCall system establishes a voice connection directly to a call center initiated either manually by vehicle occupants or automatically via activation of in-vehicle sensors. At the same time, actual location, available incident, or medical data will be sent to the eCall operator receiving the voice call. To address privacy issues and prevent potential misuse, an eCall system requires mechanisms for secure authorization and confidential transmission.

(ii) *Location-based information*: location-based information services might for example allow the driver to find the nearest business of a certain type, for example, the next fuelling station, the next ATM, or accommodations and restaurants available in the immediate vicinity. Optionally, the driver might allow certain location-based incoming information such as traffic news, local objects of interest, or localized advertisements. To prevent potential misuse, we need a secure authentication and authorization for all incoming messages. Outgoing queries, however, require adequate protection of the driver's privacy.

(iii) *Location-based billing*: having the current vehicle position would also enable certain automatic vehicular billing applications, for example, for toll roads or parking. Then drivers could securely pay by a simple acknowledgment within their car while the operating company or authority would not need to maintain an expensive billing infrastructure. This scenario, however, again needs efficient and reliable cryptographic mechanisms to mutually ensure payments while protecting the driver's privacy. Furthermore, only secure positioning could reliably enable advanced applications such as restricted areas of operation or upcoming pay-as-you-drive insurances.

(iv) *Fleet management*: modern fleet management systems enable, in addition to vehicle tracking, advanced functionality such as centrally managed routing and efficient dispatch, driver authentication, remote diagnosis while gathering details on current driver's status, mileage, fuel consumption, or container status. Therefore, a fleet management system demands for mechanisms to establish secure connections to the

vehicle's onboard computer and requires appropriate mechanisms to provide in-vehicle driver authentication and authorization.

6.7. Legal authority support

Support for various vehicular legal authority applications could become a crucial impulse for automotive electronics. Since official applications very often involve sensitive personal information, they often demand appropriate IT security measures to become enabled. In the following, some feasible applications for legal authority support in the automotive area are listed.

(i) *Electronic road signs*: there already exist developments [40] for dashboards with integrated traffic sign recognition systems that will warn the driver whether he is driving too fast. Since present traffic sign recognition systems still process digital images of their environment, future approaches will also integrate electronic road signs that wirelessly transmit their (variable) information about actual speed limits, road works, traffic jams, or road conditions. However, to detect bogus or faked information, the vehicle requires an appropriate IT security architecture to reliably verify incoming traffic sign information for validity.

(ii) *Electronic license plate*: integrating a wireless transponder into a vehicle that broadcasts an (unique) identification string will be another promising automotive development. Such an electronic license plate could for instance help to easily implement tolling and payment systems, or particularly help police forces and public authorities to identity a vehicle in case of accident or law violation. However, an adversary could modify or just steal an electronic license plate for misinformation or impersonation. Drivers in turn, require that toll road stations or an arbitrary road user cannot acquire the same amount of information as for instance qualified police forces. Thus, the application of electronic license obviously requires an adequate vehicular IT security architecture that regards attackers from both outside and inside.

(iii) *Electronic log books*: providing evidence for accomplished trips or critical maintenance operations can be very important for legal restraints, commuting accounts, or warranty claims. Having an integrated electronic service check book and/or driver's log would clearly ease bookkeeping and provide reliable information. However, both demand appropriate manipulation and privacy protection.

7. CONCLUSION: CHALLENGES AND OPPORTUNITIES FOR THE AUTOMOTIVE IT COMMUNITY

In this contribution, we presented a state-of-the-art overview of IT security in vehicles. After a short introduction to cryptographic terminology and functionality, we identified the need for automotive IT security while presenting the specific attackers and attacks within the automotive area. We introduced core security technologies and relevant security mechanisms required to protect current and future vehicular applications, business models, and components that rely on IT security. In summary, it can be stated that embedding IT security in vehicles.

(1) protects against manipulations by outsiders, owners, and maintenance personnel;
(2) increases the safety and reliability of a vehicular system;
(3) enables new IT-based automotive applications and business models.

As sketched above, there are several difficulties to overcome in order to develop strong embedded security solutions. We would like to give an outlook on the future of IT security in cars in the form of the following recommendations and conclusions.

(i) IT security will be a necessary requirement for many future automotive applications.
(ii) IT security will allow a multitude of new IT-based business models, for example, location-based services or fee-based flashing. For such systems, security will be an enabling technology.
(iii) IT security will be integrated invisibly in embedded devices. Embedded security technologies will be a field in which manufacturers and part suppliers need to develop expertise.
(iv) IT security solutions have to be designed extremely carefully. A single "minor" flaw in the system design can render the entire solution insecure. This is quite different from engineering most other technical systems: a single nonoptimum component usually does not invalidate the entire system. An example is the content scrambling system (CSS) for DVD content protection, which was broken easily once it was reverse-engineered.
(v) Embedded security in vehicles has to deal with very specific boundary conditions: computationally and memory-constrained processors, tight cost requirements, and physical security.
(vi) The multitier manufacturing chain for modern vehicles (OEM and possibly several layers of suppliers) can have implications for the security design. It is, for instance, relevant who designs a security architecture and, most importantly, who has control over the cryptographic keys.
(vii) Merging the automotive IT and the embedded security community will allow many new applications. However, there are also several challenges: security and cryptography have historically been a field dominated by theoreticians, whereas the automotive IT is usually done by engineers. The culture in those two communities is quite different at times, and both sides have to put effort into understanding each other's way of thinking and communicating.

REFERENCES

[1] A. Saad and U. Weinmann, "Automotive software engineering and concepts," in *GI Jahrestagung*, pp. 318–319, Frankfurt, Germany, September-October 2003.

[2] E. Nickel, "IBM automotive software foundry," in *Press Conference on Computer Science in Automotive Industry*, Frankfurt University, Frankfurt, Germany, September 2003.

[3] ISO/IEC, "Information technology—guidelines for the management of IT security—part 1: concepts and models for IT security," Tech. Rep. TR 13335-1, ISO/IEC, Genf, Switzerland, 1996.

[4] R. Shirley, "Internet security glossary," Tech. Rep. RFC 2828, GTE/BBN Technologies, Cambridge, Mass, USA, May 2000, http://www.rfc-editor.org/rfc/rfc2828.txt.

[5] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley, Reading, Mass, USA, 2003.

[6] W. Stallings, *Cryptography and Network Security*, Prentice-Hall, Englewood Cliffs, NJ, USA, 4th edition, 2005.

[7] National Institute of Standards & Technology, "FIPS-46-3: Data Encryption Standard (DES)," October 1977, reaffirmed in October 1999.

[8] National Institute of Standards & Technology, "FIPS-197: Specification for the Advanced Encryption Standard (AES)," November 2001.

[9] J. Daemen and V. Rijmen, "AES proposal: rijndael," in *Proceedings of the 1st Advanced Encryption Standard (AES) Candidate Conference*, Ventura, Calif, USA, August 1998.

[10] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Journal of the American Institute of Electrical Engineers*, vol. 55, pp. 109–115, 1926.

[11] C. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[12] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[14] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[15] V. Miller, "Uses of elliptic curves in cryptography," in *Advances in Cryptology (Crypto '85)*, H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Scienc*, pp. 417–426, Springer, Berlin, Germany, 1986.

[16] IEEE P1363-2000, "Standard Specifications for Public Key Cryptography," http://standards.ieee.org/catalog/olis/busarch.html.

[17] T. Miehling, B. Kuhls, H. Kober, H. Chodura, and M. Heitmann, "Security module specification," Tech. Rep., HIS-Herstellerinitiative Software, Bochum, Germany, July 2006, Version 1.1.

[18] IEEE 1609.2-2006, "Trial-Use Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages," http://ieeexplore.ieee.org/servlet/opac?punumber=11000.

[19] R. Rivest, "RFC 1321: the MD5 message-digest algorithm," April 1992, http://www.ietf.org/rfc/rfc1321.txt.

[20] National Institute of Standards & Technology, "FIPS-180-2: secure hash standard (SHS)," August 2002.

[21] U.S. Department of State, International traffic in arms regulations (ITAR), code of federal regulations, title 22, parts 120–130.

[22] P. van Oorschot, "Revisiting software protection," in *Proceedings of the 6th International Conference on Information Security (ISC '03)*, vol. 2851 of *Lecture Notes in Computer Science*, pp. 1–13, Bristol, UK, October 2003.

[23] S. Amendola, "Improving automotive security by evaluation—from security health check to common criteria," Tech. Rep., Security Research & Consulting GmbH, Bochum, Germany, 2004.

[24] A. Weimerskirch, C. Paar, and M. Wolf, "Cryptographic component identification: enabler for secure vehicles," in *Proceedings of the 62nd IEEE Vehicular Technology Conference (VTC '05)*, pp. 1227–1231, Dallas, Tex, USA, September 2005.

[25] C. Linn and S. Debray, "Obfuscation of executable code to improve resistance to static disassembly," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 290–299, Washington, DC, USA, October 2003.

[26] C. S. Collberg and C. Thomborson, "Watermarking, tamper-proofing, and obfuscation—tools for software protection," *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 735–746, 2002.

[27] J.-P. Hubaux, S. Čapkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy Magazine*, vol. 2, no. 3, pp. 49–55, 2004.

[28] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Proceedings of Embedded Security in Cars Workshop (ESCAR '04)*, Bochum, Germany, November 2004.

[29] Car-2-Car Communication Consortium. http://www.car-2-car.org/.

[30] Network on Wheels, http://www.network-on-wheels.de/.

[31] CVIS—Cooperative Vehicle-Infrastructure Systems. http://www.cvisproject.org/.

[32] Safespot, Cooperative vehicles and road infrastructure for road safety, http://www.safespot-eu.org/.

[33] K. Lemke, A.-R. Sadeghi, and C. Stüble, "An open approach for designing secure electronic immobilizers," in *Proceedings of the 1st International Conference on Information Security Practice and Experience (ISPEC '05)*, pp. 230–242, Singapore, April 2005.

[34] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1917–1928, Miami, Fla, USA, March 2005.

[35] European Commission. EU NO 1360/2002, June 2002, Corrigendum to commission regulation adapting for the seventh time to technical progress council regulation (EEC) no 3821/85 on recording equipment in road transport.

[36] Gieschen Consultancy, Report: IP theft up 22%, massive $3 trillion counterfeits, May 2005, http://www.bascap.com/.

[37] R. J. Anderson, "On the security of digital tachographs," in *Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS '98)*, pp. 111–125, Springer, Louvain-la-Neuve, Belgium, September 1998.

[38] S. Ross, "Parts counterfeiting," October 2004, http://www.aftermarketbusiness.com/aftermarketbusiness/article/articleDetail.jsp?id=125346.

[39] eCall Driving Group, http://ec.europa.eu/information_society/activities/esafety/forum/ecall/index_en.htm.

[40] Siemens VDO, Traffic sign recognition. http://www.siemensvdo.com/products_solutions/cars/propilot/.