Embedded Security Solutions for Automotive Applications

A. Bogdanov, D. Carluccio, A. Weimerskirch, T. Wollinger, escrypt GmbH – Embedded Security

Abstract

In this paper we present a number of architectural security solutions based upon concrete hardware components such as customized security controllers, trusted platform modules (TPMs), "security boxes", FPGAs and ASICs. We analyze benefits and disadvantages of each solution proposed in terms of physical and cryptographic security, costs, needed and achievable performance. We also discuss the consequences of the solutions with respect to several wide-spread security applications including immobilizer systems, component identification, software flashing, etc.

1   Overview

During the last years the significance of information security in embedded systems has drastically increased. This is due to the fact that embedded systems are getting more and more pervasive. In the context of their growing networking degree embedded security becomes particularly important.

One of the most rapidly developing areas where information security plays a considerable role is the automotive industry. A modern automobile has up to 80 embedded microcontrollers (control units - CUs) on board which control most processes in the car (e.g., engine control, steering and braking systems, navigation systems, traffic control, etc.) and build a heterogeneous, multi-rank communication network with broadband interfaces to open computational environments (see Fig. 1). The cost for electronics and software is approaching 30% of all manufacturing costs.

Security solutions based on software approaches have been gaining attention in the automotive industry throughout the last years. These are vulnerable to attacks based on malicious software, though. Recently a discussion about a secure hardware component as security anchor has been started. However, it is unclear yet how secure computing modules are able to add security to a vehicle, and which security solution should be preferred. In this white paper we give some initial thoughts about these aspects in order to start a discussion about future approaches.

2   Secure Hardware Computing Base

Providing security in embedded systems is a challenging task requiring an interdisciplinary approach encompassing cryptographic methods, technical security solutions and organizational measures which should be taken to provide an adequate security level. Traditionally, the majority of high-security solutions in the embedded area are based on one of the following secure computing platforms.

2.1 Security Controllers

Security controllers are special microprocessors protected against active (tampering and other invasive attacks) and passive (timing attacks, simple differential power analysis, internal collision attacks, EM analysis, template attacks and many others) physical attacks. They offer a number of pre-implemented cryptographic services such as DES/3DES, AES, hash functions, finite field and residue ring arithmetic, RSA, secure generation of random numbers (needed for key management, some signature schemes, security protocols, etc.) and others. These cryptographic functionalities are usually implemented as co-processors. Such controllers are also able to store data in secure area. Hence, usually the data can be written once, but can afterwards only be read out or only used by the security controller for cryptographic operations. Most of these controllers are smart card derivations delivered in

traditional microcontroller packages (e.g., DIL, TSSOP, DSO, etc.). Such security controllers possess 8-, 16- or 32-bit central processing units with clock frequencies between 8 and 66 MHz, 2-16 Kbyte RAM, 16-256 Kbyte ROM, and up to 400 Kbyte EEPROM. There are, however, security controllers with larger (up to several Mbytes) EEPROM and ROM.
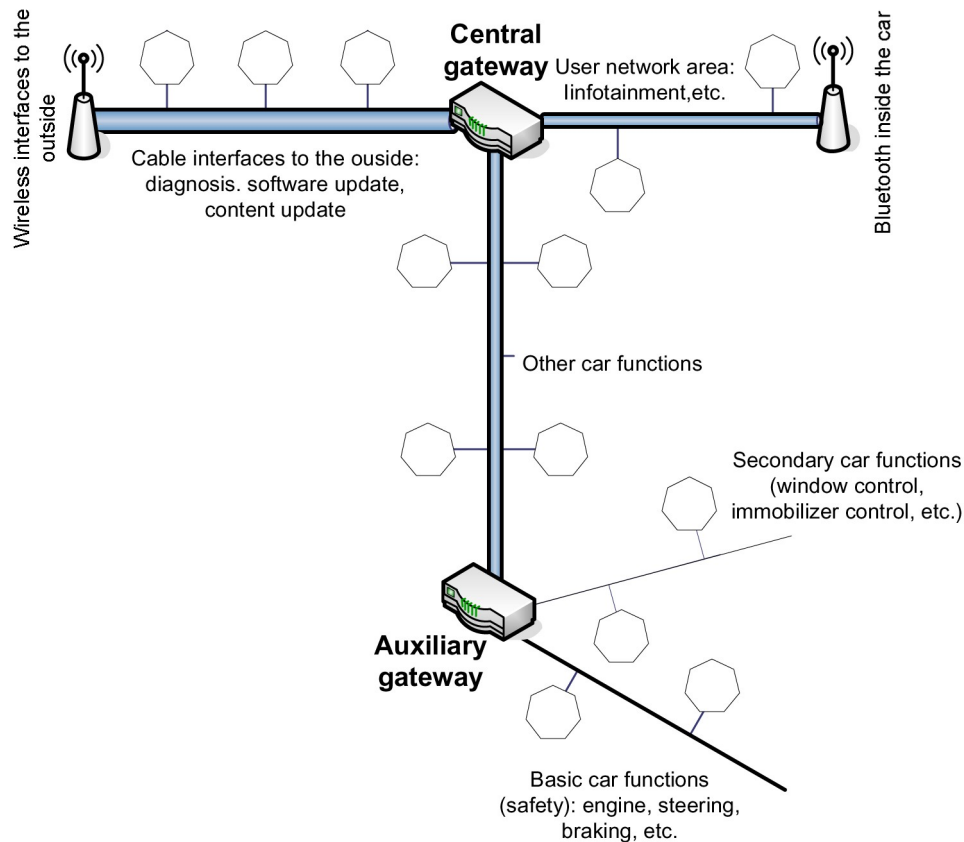

Fig. 1. General in-vehicle network topology

The advantages of such controllers are as follows:

u   These are special-purpose high-security solutions whose hardware and firmware architectures are well-understood and whose security has been as a rule thoroughly evaluated and certified by state certification bodies within formalized certification procedures (e.g., by Common Criteria (CC) [2] Protection Profiles (PP) [3], [4];

u   They are already available on the market and can be relatively quickly produced after developing the corresponding embedded software application;

u   Relatively low manufacturing costs (their price is in the range of one to several Euros for high volumes).

At the same time there are several technical drawbacks impeding their potential wide adoption in automotive applications:

u   Relatively low computational performance in view of the need of real-time reaction in safety-relevant applications;

u   Relatively low data transmitting capabilities which may prevent standard security controllers from controlling broad-band in-car communication and external interfaces online;

u   Relatively narrow range of operational conditions. E.g., the allowed temperature range is as a rule from -20 °C to +85 °C which is to be compared to the required values for automotive applications from -40 °C to +105 °C. For solving these problems some modifications in the hardware core through semiconductor manufacturers may be necessary.

2.2 Trusted Platform Modules

A special type of smart-card based security controllers are so called Trusted Platform Modules (TPMs) whose security functionality is defined by the TCG (Trusted Computing Group) standardization body in [5]. Originally TPMs were developed to provide platform integrity (software and hardware integrity) of PCs with emphasis on software attacks. The security services provided by TPMs and their interfaces are standardized.

The basic functionalities of a TPM are the following:

- Secure storage of cryptographic keys and hash values (representing platform configuration);

- On-chip key pair generation using an unpredictable hardware-assisted random number generator;

- Computation of hash-values;

- Signature generation and verification;

- Monotonic counter.

TPMs have clear advantages:

- These are ready products wide-spread on the information security market;

- As a rule they are CC certified by their manufacturers;

- TPMs use clearly specified, well-understood and transparent security mechanisms;

- They are relatively cheap (in the range of one Euro).

But their drawbacks are much more considerable:

- TPMs offer a limited set of cryptographic services which is often not sufficient to build a multi-functional information security system apparently required in the automotive area;

- The available functionality is fixed by the TCG and chip manufacturers and cannot be extended;

- To securely use their functionality it is almost always necessary to have a trusted computation environment (or some elements of it) outside the TPM;

- Moreover, all limitations of security controllers mentioned in Section 2.1 persist.

2.3 "Security Boxes"

"Security boxes" are high-performance computer devices additionally protected against physical attacks (tamper and side-channel resistance). They can use standard high-end microprocessors as well as bespoke FPGAs/ASICs and possess a performance level comparable to modern PCs and higher. Security boxes are deeply customized products and available at the moment mainly in military and state security applications (e.g., communication devices installed in embassies, army headquarters, fighter aircrafts, and anti-aircraft defence emplacements). The concept of security boxes found limited application in some industries as well (e.g., measurement in the trunk distribution of electricity and energy carriers).

Secure boxes have a number of benefits valuable for the automotive industry. The most important ones are:

- High computation (almost arbitrary hardware base can be taken) and communication (among others because of the possibility of hardware-assisted realization of communication protocols) performance;

- High performance of cryptographic algorithms, since they can be realized in hardware which is as a rule much more efficient;

- High storage capacity.

All these properties enable security boxes to perform centralized online control of automotive security features.

The only relative disadvantage of this approach is that there is neither a solid development base nor ready solutions available. That is, security boxes must be developed from the outset, which means high development costs. The same applies to certification procedures (e.g., there exists no Common Criteria protection profile for automotive security boxes). Moreover, security boxes will be relatively costly in manufacturing.

3   Security Objectives

The security issues in vehicles usually comprise the following topics:

- Electronic immobilizer;
- Remote entry system;
- Secure software download (ensure software integrity);
- Secure access (e.g., for diagnosis purposes and further external channels);
- Digital rights management (e.g., infotainment);
- Mileage counter manipulation;
- Component identification (ensure hardware integrity, e.g. detects theft and forgery).

Some examples for future scenarios including security are:

- Car-to-car communication;
- Car-to-infrastructure communication;
- Data event record;
- X-by-wire;
- Speed control;
- Toll collection;
- Secure internal communication;
- And so on.

Concerning data, software and hardware manipulation there are two security levels to achieve:

1. **Avoiding manipulation:** Every state of the system is permanently controlled by the security mechanism in real time which guarantees that the system is in a secure state at any time moment.

2. **Detecting manipulation:** If the system has been once in an insecure state, the security mechanism will detect that within a finite number of steps after this event. Practically, we are interested in a security mechanism detecting manipulation directly at the time of manipulation or immediately after this.

While it is desirable to secure crucial vehicle components such as the engine this seems to be out of scope today. Hence, we focus on securing the CUs connected to an internal bus. The overall security goals can easily be summarized to avoid and detect manipulation, respectively, in order to achieve software and hardware integrity. Concerning data confidentiality, one can speak of reliably providing confidentiality or failing to provide it. Moreover, similar classes of avoiding and detecting information leakage can be introduced.

4   Security Architectures

There are three basic security architectures:

- a centralized approach where all security is handled in a centralized way (client-server architecture),

u   a distributed approach (peer-to-peer architecture),

u   a semi-centralized approach which is a combination of the centralized and distributed architectures.

However, each concrete approach must be mapped to hardware solutions (standard controllers, TPMs, customized security controllers, security boxes).

4.1 Centralized Architecture

In the centralized approach there is a single security module that is responsible for the overall security. This module has to withstand State-of-the-Art attacks and has to perform the necessary security operation in an acceptable speed. However, the required performance depends on software architecture and is application specific.

Each CU shares a relationship with the security module (e.g., an individual secret key). The security system can in this case realize a server-based authentication and key-establishment protocol, the security module (SM) being the protocol trusty. If a CU_i wants to build a secure channel to CU_j, it asks SM to participate in establishing a session key between these two CUs. After establishing a key, the both parties can communicate with each other not involving SM. The CUs in this case have neither to maintain all cryptographic keys of their communication partners nor to store any master keys in their potentially insecure computational environment. This considerably reduces the computational power required of SM to maintain in-car communication sessions. There is, however, still the necessity to be able to distribute all necessary keys fast enough to support real-time (e.g., at the start-up time) and safety critical (e.g., braking) applications. The SM has to store only a single car master key. It can derive keys needed to communicate with CUs from this single key and the key storage requirements can be significantly reduced. However, the communication to the outside should be performed by SM directly without delegating this to a less protected CU. In this way all out-car communication sessions can be recorded. Moreover, the SM can also incorporate several security relevant CUs such as electronic immobilizer, security relevant parts of the engine control unit and mileage counter.

The security module might be a dedicated CU or it might be part of a CU that already exists (such as the head unit, engine control unit or central gateway). There might be a communication channel between the security module and an external entity, e.g., the OEM or a mechanics via a diagnosis channel. This architecture is depictured in Fig. 2.
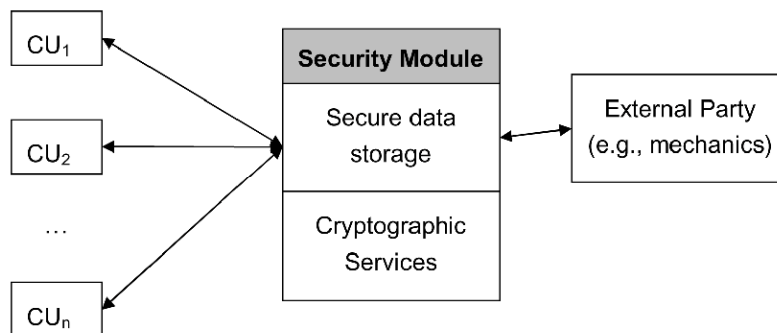

Fig. 2. Centralized Architecture

Note that the security module is the central point of failure and attack. Hence, compromising the security module must be at least as hard as compromising all (or a considerable part of) CUs. If S denotes the security level, then it must hold: $S(SM) \geq S(CU_1) + S(CU_2) + \ldots + S(CU_n)$, where n equals to the number of CUs in the car.

In the centralized architecture the security module may be realized on the following bases:

u   *Standard non-security processor:* This solution can be acceptable in rather unlikely cases where physical attacks are no concern (e.g., if malicious physical access to the car is securely protected by organizational measures). Even in this case the processor has to be powerful enough to maintain key establishment and fast communication to the outside.

u   *Security controller*: This can be a solution of choice, provided the security controller is made robust to the automotive environment and supplied with fast and secure cryptographic co-processor to maintain high-speed out-car communication.

u   *Security box:* This hardware solution seems to be the most suitable one for the purpose of central security unit. It can undoubtedly provide both real-time and out-car protected data transmission and control.

4.2 Centralized Architecture

The distributed architecture does not know a central security module but distributes the role of the single security module. The distributed architecture can be realized in a way such that the CUs manage to distribute the security functionality in an autonomous manner and appear to the external unit as one security module. A different approach could be that the external party provides management functionality and queries each CU. Again, each CU requires a security client either in software or hardware. The CUs might be related to each other, e.g., by shared secret keys or they might be related to the external party.

In the easy case the external party shares a secret key with each CU. Security is then based on this relationship and the possibility of the external party to communicate with each CU. This would mean high storage complexity for each individual CU and the limitation of the security of the whole system to the security of the weakest CU (or a set of the weakest CUs needed), since each CU must then contain all keys it needs. Otherwise the key logistics would be to complex. In the second case all distributed CUs provide an abstract single security module to the external party. In such case the key management comes at far lower overhead.

However, such an approach needs to be more researched since several aspects of its implementation are unclear today. The distributed architecture is depictured in Fig. 3.
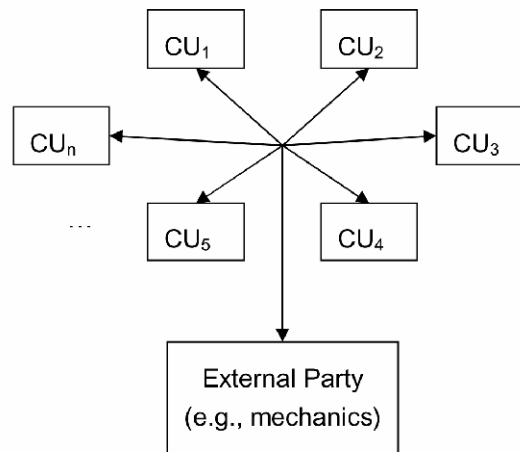
Fig. 3. Distributed architecture

As to possible concrete hardware realizations, the following solutions seem to be most appropriate:

u   *Standard non-security processors*: In this case system designers have to come up with a distributed key establishment system possessing suitable performance for real-time and safety critical applications.

u   *Several security controllers*: For the most security demanding applications several security controllers are used, the other CUs being based on standard non-security processors.

u   *Several TPMs:* Some applications requiring only limited security functionality can be realized using TPMs as well. A TPM is, however, dependent on its owner that calls its functions from the TPM and needs to be implemented in a secure environment.

4.3 Semi-Centralized Architecture

The semi-centralized architecture incorporates security features of both centralized and distributed approaches: There is a central security module supporting security functionality of most CUs; at the same time, some of the CUs (probably, the most security demanding ones) are realized as separate security modules with their own communication channels to the outside world, see Fig. 4. These separate security modules might have connections to the central security module and/or to the CUs over their own key establishment system (this would increase the key storage complexity in each CU only linearly). Potential candidates for the separate security modules are engine control unit, electronic immobilizer, and mileage counter.
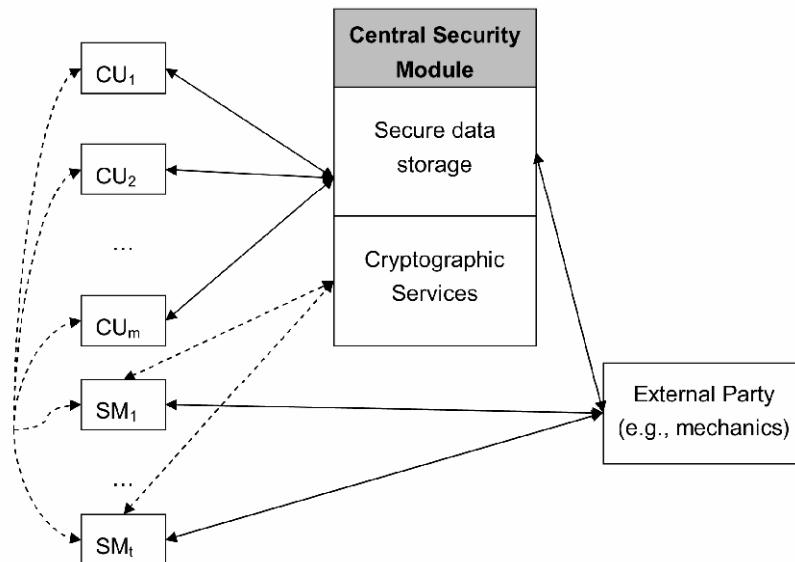
Fig. 4. Semi-centralized architecture

The most reasonable hardware solutions for the semi-centralized architecture include:

u *Standard non-security processors (CUs), security controllers (SMs) and a security box (CSM)*: Applications that do not require a high security level (e.g., windshield wiping controller, sitting controller, etc.) can be implemented on standard microprocessors. The corresponding software should however support basic security functionality. Some security relevant applications can be implemented in security controllers which can delegate computationally intensive tasks to the central security module based on a security box.

u *Standard non-security processors (CUs), security controllers (SMs/CSM)*: The communication between individual CUs (standard non-security processors) and SMs (security controllers) is controlled by a central security module which is based on a security controller. The CUs can also use some security functionality of the central security controller.

5   Security Analysis

In the following we will analyze above architectures in more detail. As already mentioned the autonomous distributed approach requires further research such that we believe only the centralized architecture and the simple distributed approach to be deployed in the near future.

It seems clear that any economically rational security strategy is not able to *prevent* an attack based on hardware manipulation. For instance, it is always possible to replace crucial CUs by malicious components that provide the exact same functionality without the security functions. A proper example is a CU that does not verify the electronic immobilizer. The major security objectives one can achieve are as follows:

1. *Avoid* manipulation due to attacks based on malicious software,

2. *Detect* attacks based on manipulation of hardware.

The first aspect can be ensured by defining a proper interface from the outside world to the vehicle and by protecting this interface. For example, standard access control and authentication of any data imported to or exported from the vehicle will allow such an approach. Furthermore, for critical components such as the engine control unit the standard interfaces can be protected by cryptographic means. For instance, the flash boot loader should be appropriately protected. However, protecting such devices of direct hardware manipulation, e.g., by direct physical contact, is only possible to a certain point. It is possible to protect the flash memory of direct contacting and manipulation by using a dedicated security chip. However, it seems almost impossible to prevent an attack where an attacker just introduces a further controller that partly takes over the role of the old one. For preventing attacks based on software there is no need for a dedicated hardware security module that incorporates features of a security chip but the functionality might be simply implemented in software on a standard controller. This goal can be achieved with all architectures in question (centralized, semi-centralized, and distributed). The main issue to consider here are the interfaces to the external world and how they are protected.

The second major goal of detecting a hardware manipulation can be achieved at a high probability but must be thoroughly planned. The main idea is as follows:

- *Centralized and semi-centralized security architectures*: The central security module queries the CUs at determined events and records all responses. It applies plausibility checks to the results and is able to detect manipulations. The security module might directly react to a security breach or notify an external party (e.g., once a diagnosis check is performed).

- *Distributed security architecture*: Each CU records its events. An external entity (say, an OEM trusted mechanics) is able to read out all these records, e.g., at the time of a diagnosis check. The external party than applies plausibility checks and is able to detect irregular behavior at a high probability.

The plausibility checks depend on the characteristics of the CU. For instance, the following plausibility checks might be applied for different CUs:

- *Mileage counter:* The security module can store a copy of the mileage counter or it can store the actual mileage counter. Thus, the sensors would send the signal to the security module which then increases the mileage counter appropriately. If the security module has secure storage the counter cannot be altered once it is stored in the module. An attacker would have to alter the communication to the security module. However, the module could check whether the signal is disrupted (e.g., by checking whether the car is turned on for a long time without any sensed driving) or if it is altered (e.g., by checking whether the car is always going very slow). A trusted third party such as the OEM could then read out the records and detect the manipulation.

- *Electronic immobilizer:* The role of the electronic immobilizer is to ensure that a vehicle can only be turned on if the appropriate vehicle key was used and if the appropriate CUs are present in the car. Usually, if the check is successful the engine control unit is activated and the engine can be started. An easy attack is to just replace the engine control unit and, if necessary, further crucial CUs. The security module could record the responses of the security check when querying all involved CUs. If the CU was replaced, there would be no proper response of the key and the engine control unit, but the security module would detect that the vehicle is still running. A trusted third party could then detect the attack.

- *Component identification:* Component Identification (also see [1]) ensures hardware integrity of the vehicle. It is in general very similar to the mechanisms of the electronic immobilizer (which verifies hardware integrity of the key and dedicated CUs). Hence, the same mechanisms can be used here.

A common attack will be to remove or replace the security module, or to add a second module that reacts to the other CUs in the same way. However, with a plausibility check and the knowledge of the secret keys of the security module, a trusted third party is able to detect such a manipulation.

Note that detection of hardware manipulation can only be detected up to a certain degree. An adversary that is aware of how the plausibility check is performed and that has sufficient monetary resources is always able to compromise the system. However, we believe that it is possible to design plausibility checks in such a way that an attack's effort is higher than the gain such that no rational adversary will implement such an attack.

We started this article with a description of TPMs. But how well suited are TPMs for the application in vehicles?

The pros and cons of a TPM to a custom specific solution (e.g., an ASIC or an FPGA solution) are summarized in the following table:

| | TPM | Custom specific | |
		ASIC	FPGA
Standardized	Yes	No	No
Flexibility	No	Yes	Yes
Price	Medium	Low for high volumes	High
Security level	High	Adaptable to required level	Medium - high

Finally, one can say that there is no winner. TPMs have the clear advantage of being a standardized security module with high volumes. On the other side, they come with an interface that was designed for the PC world. There are approaches to adapt TPMs to the mobile and vehicle world though. A custom specific solution can be exactly suited to the requirements of vehicle OEMs regarding functionality and required security level. However, an appealing prize seems only possible if several OEMs agree on a standard security module such that high volumes can be reached. An FPGA solution seems only reasonable for small volumes due to its prize.

6 Outlook

In this report we analyzed using a security module in vehicles. Clearly, several issues need to be resolved first. For instance, the security module needs to hold the stress and temperature requirements in vehicles. However, the chances for a security module are appealing:

- u A single security module might safe code size and hence even cost, because one needs less memory, processor power as well as developing cost.

- u Software attacks can be prevented and hardware attacks can be detected, hence software as well as hardware integrity can be proven by a trusted third party.

- u Communication to the vehicle can be protected by the security module (e.g., as part of a central communication gateway for diagnosis, wireless communication as well as further channels) such that introduction of malicious communication is detected and has no affect.

- u Business models requiring a secure digital rights management (DRM) can be introduced (e.g., aftermarket feature activation and custom specific use of digital contents).

However, the main advantage of a hardware security module compared to a software solution is that hardware attacks can be detected and that a dedicated security module is available that is able to perform cryptographic operations much faster and with little resources even compared to powerful 32-bit standard controllers such that in the end cost might be reduced.

The design of a practically secure and highly functional automotive computer system resistant to all software attacks concerning confidentiality and manipulation as well as some powerful physical attacks is possible. The main candidates for security architectures and secure hardware platforms are the centralized architecture (based upon a security box or a high-end security controller with a powerful cryptographic co-processor) and semi-centralized architecture (on the basis of several high-end security controllers possibly accompanied by either a powerful cryptographic processor or a central security box).

References

[1]   André Weimerskirch, Christof Paar, and Marko Wolf, "Cryptographic Component Identification: Enabler for Secure Inter-vehicular Networks", 62nd IEEE Vehicular Technology Conference, September 25-28, 2005, Dallas, TX, USA.

[2]   Common Criteria for Information Technology Security Evaluation. Parts 1, 2 and 3. Version 2.2. January 2004

[3]   Smartcard IC Platform Protection Profile BSI-PP-0002, Version 1.0, July 2001

[4]   Protection Profile Smartcard Integrated Circuit PP/9806, Version 2.0, September 1998

[5]   TCG TPM Specification, Version 1.2, Revision 94, 29 March 2006

Andrey Bogdanov, Dario Carluccio, André Weimerskirch, Thomas Wollinger

escrypt GmbH- Embedded Security
Lise-Meitner-Allee 4, D-44801 Bochum, Germany
E-mail: {abogdanov, dcarluccio, aweimerskirch, twollinger}@escrypt.com