Cyber Security in the Automotive Domain – An Overview

2017-01-1652
Published 03/28/2017

Rolf Schneider and Andre Kohn
AUDI AG

Martin Klimke and Udo Dannebaum
Infineon Technologies AG

CITATION: Schneider, R., Kohn, A., Klimke, M., and Dannebaum, U., "Cyber Security in the Automotive Domain – An Overview," SAE Technical Paper 2017-01-1652, 2017, doi:10.4271/2017-01-1652.

Abstract

Driven by the growing internet and remote connectivity of automobiles, combined with the emerging trend to automated driving, the importance of security for automotive systems is massively increasing. Although cyber security is a common part of daily routines in the traditional IT domain, necessary security mechanisms are not yet widely applied in the vehicles. At first glance, this may not appear to be a problem as there are lots of solutions from other domains, which potentially could be re-used. But substantial differences compared to an automotive environment have to be taken into account, drastically reducing the possibilities for simple reuse. Our contribution is to address automotive electronics engineers who are confronted with security requirements. Therefore, it will firstly provide some basic knowledge about IT security and subsequently present a selection of automotive specific security use cases.

Introduction

Connectivity in the automotive industry enables an increasing number of use cases and is fostering new business opportunities for OEMs (See Figure 1).

On one hand, today's customers expect all the applications they got used to on their smartphones to be available in a modern car, along with possibilities to connect personal mobile devices to it. On the other hand, new features and functionalities, like automated driving, unlocking via smart phone, navigation with detailed satellite graphics, etc., increase the need to connect the car to the outside to enable them or at least improve performance and user experience.

But when connecting the car to other devices, IT- security becomes a priority, because simultaneously the car becomes an attractive target for attackers. Therefore, confidentiality, integrity and authenticity must be maintained and additionally privacy protection will be a concern.

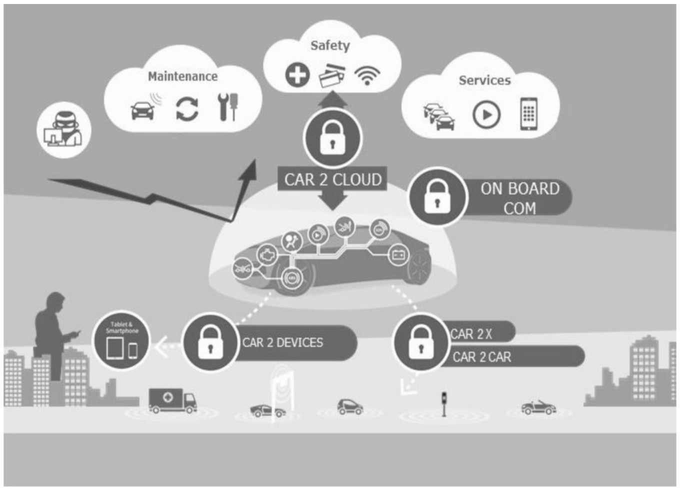


Figure 1. Connected car and the need for security

However, these topics are not common practice for an automotive electronics development engineer up to now. Therefore this paper gives an overview on some security fundamentals and will address a couple of recent working topics related to automotive security.

Fundamentals of Security

The main goal of all security measures is to maintain confidentiality, integrity and availability (Figure 2). The various communicating entities - e.g. a car communicating to the OEM backend but also the ECUs within the car communicating to other ECUs - require that they can determine the authenticity of other entities, that the data is not altered and also in some cases that it cannot be eavesdropped thus can be kept confidential.

Confidentiality
Integrity
Authenticity

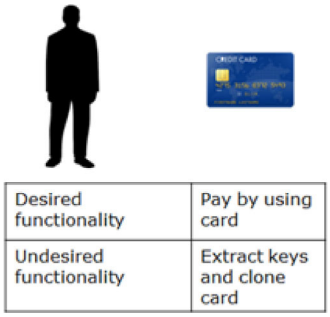
Figure 2. Security goals

The technical fundaments are cryptographic algorithms using secret keys. The secrecy of keys is critical for security. Once the integrity of keys is compromised, security is broken. With respect of potential disaster recovery, compromised keys resemble a kind of worst case scenario. An attacker who knows the keys that are used to protect the exchange of keys can eavesdrop the related communication which makes this process not applicable in unsecured environments (e.g. on the road) anymore.

Therefore affected cars have to be brought in a secured environment resulting in high effort and long process times for both the OEM and the customer. This leaves the fleet of an OEM unprotected against attacks during a potentially long period of time impacting both, safety and also the operation of the related services, negatively. Therefore protecting the secrecy of keys shows a high priority for security in automotive

Trust Anchors Protect Cryptographic Keys

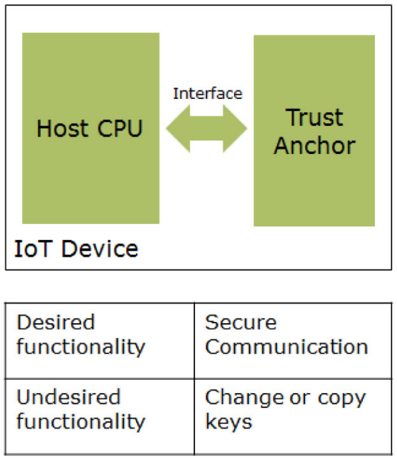
A general concept that is already commonplace in many security critical infrastructures is the usage of Trust Anchors. Trust Anchors are dedicated execution environments that host the secure storage and processing of keys. The main purpose is to separate the use of keys and encryption capabilities from the direct access to keys.



Desired functionality	Pay by using card
Undesired functionality	Extract keys and clone card

Figure 3. Payment Card controlling the rights of user

A credit card may serve as an example for a kind of trust anchor. The legitimate owner of the credit card should be able to use the card to retrieve money but he/she (or an attacker) should not be able to get hold of the keys inside card and clone the card (Figure 3). In cars, applications running on a microcontroller should be able to communicate securely but should not directly be able to read out or alter keys.



Desired functionality	Secure Communication
Undesired functionality	Change or copy keys

Figure 4. Trust Anchors connected to Host CPU

Trust Anchors can also be weakly implemented in software as part of the operating system of a microcontroller. It has been often shown that this kind of protection can be broken and cannot be considered as state of the art. The “Heartbleed bug” can serve as an example of a weakly protected key storage. This attack affected many industries and it also revealed the difficulties to recover from attacks on keys [7], [8], [9]. It showed that it is a big benefit to separate the Trust Anchor function from the overall system (Figure 4). In a way Trust Anchors mitigate the complexity problem of security by separating critical security functions from the complex overall system.

In the automotive industry this problem is understood. This resulted in the development of the SHE module [2] already in the first decade of this century, followed by the development of so called HSMs (Hardware security modules) [3] integrated into MCU like the AURIX™ of Infineon or comparable products from other vendors like NXP or Renesas.

Processing of keys is a security critical operation. Therefore in Trust Anchors special measures are implemented that harden these operations against attacks. With respect of functionality Trust Anchors have a substantially lower complexity as the overall function of an ECU they are protecting. Therefore, they can be evaluated and tested with a thoroughness which would be infeasible due to time and effort for the overall system. In a way they resemble the “divide and conquer” principle which is commonplace in engineering to cope with complexity of systems.

OEMs are starting to classify data with respect of security relevance and this classification is also affecting the measures which are required and justified for protection the related keys. In general communication that can affect the security of the whole fleet of an OEM is regarded more critical than communication affecting only one specific car. Likewise the lifetime of a key is relevant. Long lasting keys should be protected stronger than keys that will only be used for a limited time (e.g. session keys).

In the automotive industry commonly two types of Trust Anchors are deployed:

1. Trust anchors integrated silicon wise on automotive microcontrollers and
2. Trust anchors that are based on security controllers using smart card technology (e.g. Infineon OPTIGA™ TPM, NXP SmartMX and comparable).

See Figure 5 for more details.

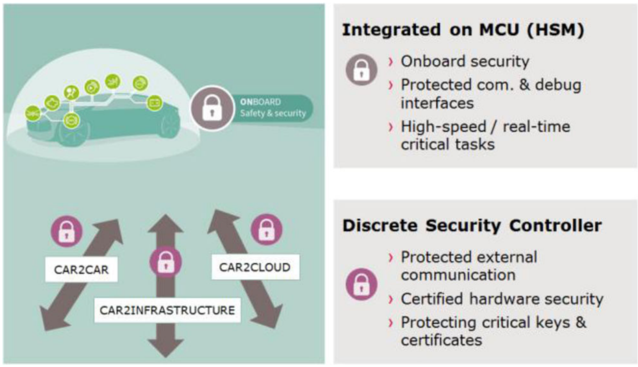

ONBOARD
Safety & security
CAR2CAR
CAR2CLOUD
CAR2INFRASTRUCTURE
Integrated on MCU (HSM)
› Onboard security
› Protected com. & debug interfaces
› High-speed / real-time critical tasks
Discrete Security Controller
› Protected external communication
› Certified hardware security
› Protecting critical keys & certificates

Figure 5. Classification of Trust Anchors used in vehicles

The main use of the first category is secure on-board communication requiring high performance and good real time ability. The second category is used to secure external communication and can also serve as a central key store for the car. In the latter category Security Controllers include comprehensive hardware measures that protect even against attacks using dedicated hardware (e.g. side channel attacks by observing power consumption or even attacks on the silicon). Additionally great care is taken to test and evaluated the critical software operations of those parts. Products are tested in many cases following a standardized security evaluation scheme called Common Criteria [10], [11] that is used widely in the smart card industry to successfully protect large, security critical infrastructures e.g. passports, signature card etc.

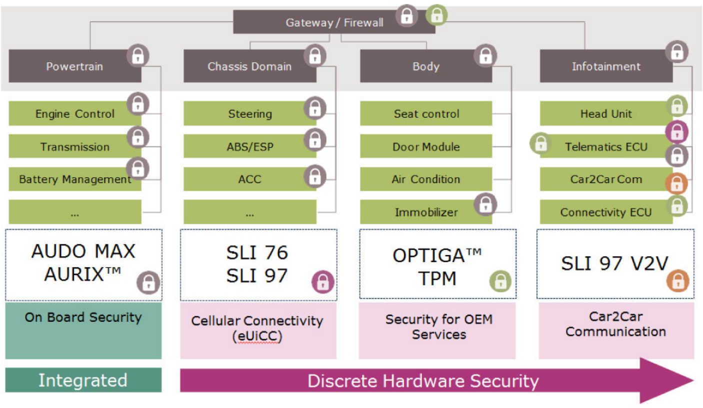


Figure 6. Automotive Trust Anchors

Aside the key storage Trust Anchors can host application specific algorithms. Vendors therefore offer different types of trust Anchors which differ with respect of performance but also security. Using Infineon's product portfolio as an example, Figure 6 shows a range of security hardware solutions for different applications.

Protection of Keys is a Result of Security Processes

Keys need protection throughout the whole life cycle of the car not only on the road but also already in manufacturing. Especially manufacturing is critical because if these devices are not properly secured they may provide an (insider) attacker an attractive attack opportunity. He/she may be able to get access to a large number of keys. Depending on the OEM specific supply chain, key injection may not only occur at one place (e.g. at the OEM) but it can be distributed at different places along the value chain. Therefore it becomes a requirement to audit and sustain the security measures to handle and process keys at all involved parties, like suppliers and contractors.

A cost efficient secure approach is the use of a personalized security controller. Personalization means that the security controller already contains a chip individual key injected in a security certified manufacturing process at the silicon manufacturer. Since security controllers are protected against hardware attacks, those parts can be shipped using inexpensive standard logistic processes without the danger of compromising the keys during shipment. The personalization process of an ECU (hosting a personalized security controller) can benefit from initial key on the security controller because further keys can be received by the ECU already protected by using secured communication. Thereby security measures in manufacturing securing the transfer of keys in plain text are not required and costs can be saved. See also Figure 8.

In general it can be shown that high availability requirements of security back-ends can be reduced and thereby overall cost can be saved when using good protected hardware trust anchors. Also the requirements to enable certain use cases e.g. like component protection in garages replacing ECUs can be lowered not only saving cost but also easing non-discriminatory service provisioning for independent garages.

These advantages show that cost considerations for security need to be done in a system wide approach including the cost of lifecycle management and fleet operation. To determine costs by solely looking at the BOM of a given ECU is therefore misleading.

Security and Complexity

Policies for Crypto Services of Trust Anchors

Protection of keys alone is not enough. Measures have to be in place that enforce that the right entity can only use crypto graphic services of trust anchor.

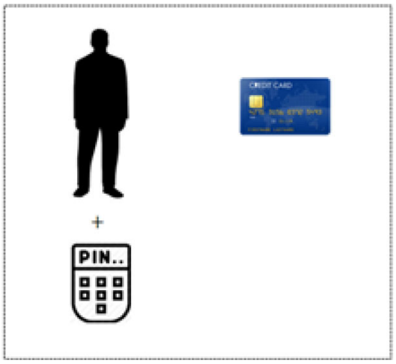
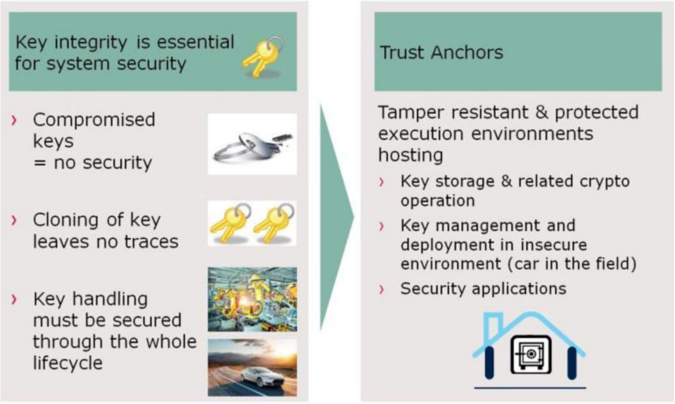


Figure 7. Policies increase security in payment

Referring to the previous example of a payment card and consider for a moment the combination of human and card as the system. It becomes clear that just the ownership of the card to perform a payment transaction would be too risky because the integrity of the human (e.g. by a thief) cannot be assured. In payment the legitimate user is provided by a pin that provides a stronger binding between payment card and human (Figure 7). In the following we will discuss how policies can be beneficially used to support security measures on the Host-CPU.

The increasing complexity of the software running on the MCUs (Microcontroller Units) within ECUs raises the risks for security flaws. Isolation and separation of software functionality is best practice that can mitigate the effect of a weakness in a certain function.



Key integrity is essential for system security

- Compromised keys = no security
- Cloning of key leaves no traces
- Key handling must be secured through the whole lifecycle

Trust Anchors

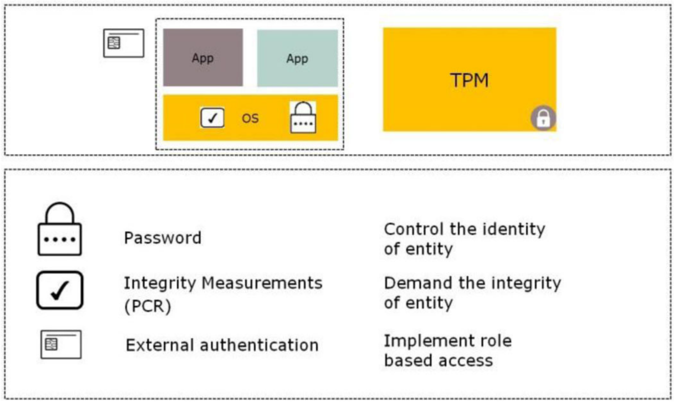
Tamper resistant & protected execution environments hosting

- Key storage & related crypto operation
- Key management and deployment in insecure environment (car in the field)
- Security applications

Figure 8. Key Security

For safety related applications it is therefore common to also separate functionality of the software executed on the MCU (e.g. separate the OS from the applications and the applications from each other). Actually the software on the MCU is not one single entity anymore but is divided into separate entities having different privileges. This approach is supported by the memory protection unit of the MCU which should not only protect against illegal write attempts but also should block against illegal read accesses. Especially protection against illegal read access is a new requirement driven by security. Therefore the MPUs of some MCU vendors already support read protection.

This isolation of functions also needs to be supported by the aforementioned trust anchors. Not every software entity on the MCU should have access to all cryptographic services residing in the trust anchor. The right to use certain keys and crypto operation needs to be assigned exclusively to certain software entities running on the MCU. In case of the TPM so called security policies can be used. The simplest approach is that an application needs to provide a password to the trust anchor to get access to a cryptographic service with a specific key. This password is stored in the data/code of the application on the host and therefore read protection of this data/code is required.



••••	Password	Control the identity of entity
✓	Integrity Measurements (PCR)	Demand the integrity of entity
▤	External authentication	Implement role based access

Figure 9. TPM access policies support Host MCU security measures

In TPM 2.0 these policies have been largely extended by various parameters that can be flexibly combined to support a great variety of use cases using a standardized approach (Figure 9).

For example aside the password, an access policy can be implemented that enforces integrity measurements of the related software wanting to use certain keys and cryptography.

But also more complex use cases can be realized easily by so called Enhanced Authentication [12]. An example is the access control to a secure diagnostic port for deep level inspection of a car, which is only allowed to be used by dedicate personal at the OEM. In this case an OEM corporate policy could enforce security by assigning the right to perform these functions to dedicated persons by issuing a hardware security token combined with a password. Since TPM is also supporting an access policy that requires an external authentication, such use case can be realized based on the TPM standards.

ECU Intrinsic Firewalls

The head unit and infotainment resemble the most complex system with respect to lines of code. Therefore it is a safe bet to assume that there will be undetected security weaknesses when the car is shipped. Direct access to the safety critical on board car busses must be avoided. Increasingly dedicated automotive microcontrollers supporting various car busses and providing good IO capabilities are used to separate the access of the infotainment CPU to the car busses. In this case, the microcontroller can act as a firewall and can also assure that potential malware of the infotainment CPU cannot compromise the safety critical functions of the onboard car communication. The HSM is instrumental in securely updating the firewall rules and also in enabling secure communication on the car busses.

Furthermore trust anchors play a vital role in the infotainment domain because they can be used to secure the lifecycle management of various services that are being hosted on the infotainment system. Since the services may evolve and also increase over the lifetime of the car, it is mandatory that the key storage has no hard limits with respect to the number of keys it can handle. A TPM therefore supports a standardized secure key storage that can be physically stored in the memory of the host CPU. This means the number of keys is not limited by its internal nonvolatile memory (e.g. Flash).

Detect Early and Patch Fast

The security concepts of vehicles also need to consider the detection and mitigation of security incidents.

Various technologies are considered to be used in the car:

1. Intrusion Detection Systems (IDS) hosted in central or domain gateways to detect abnormal behavior and report deviation to an OEM Backend and the related OEM cyber security response teams.
2. Trust anchors which are protecting asymmetric keys are used for the remote device management. This allows fast security patches in the field without the need for a garage stop.
3. ECUs that can perform self-healing to recover to a non-compromised state from a protected backup storage. Self-healing can be a reaction of an ECU after it has detected an integrity deviation for its software e.g. during a secure booting procedure or also during normal operation. Once the car is in a safe state, the ECU can request an authentic software backup from a central repository located in the car.
4. Stop service and inform the driver.
   In some cases, especially during attacks, which are using external communication interfaces, a prompt reaction may also be to temporarily stop an affected service and inform the driver. This is obviously not a general concept and can only be applied if availability requirements are not hurt.

Automotive specific Challenges

Reuse of established and widely used algorithms and procedures for security mechanisms/implementations has both, the benefit of lowering the effort but also increasing the security. After having experienced the downside of vendor specific, proprietary algorithms [13], [14], the car industry is in agreement to use standardized crypto algorithms e.g. AES, RSA and ECC and that are publicized and thoroughly reviewed by scientists in a public discourse. But not only cryptographic primitives benefit from standardization, also more and more standardized security protocols like TLS are used. This approach is also supported by TPM. It does not only provide standardized functions implemented in the TPM but also specifies the functionality that is required on the host CPU. This technology has been in use for more than 14 years in PC and servers. Car security can benefit from this security experience.

Although reuse of existing and proven security technology is desirable, the car has other specific requirements that need to be taken into consideration.

Environmental conditions

In general cars need to fulfill a high quality level and should endure robust environment conditions. Both must be reflected also in the security critical parts. Especially smart card technology deployed in the car needs to support extended temperature range and requires an automotive specific security qualification and manufacturing process. The first strongly affected security controller was the SIM (mobile subscription module). The car industry required a solderable SIM to cope with vibrations, extended temperature range and an automotive quality level (AECQ-100). As an industrial example, the SLI 76 from Infineon could be mentioned. Comparable secure elements are available from other semiconductor vendors, too.

Legacy busses and ECUs

Current car platforms are renewed every 5-7 years. A new car platform does not necessarily mean that every ECU in this platform is developed from scratch or all car communication busses are updated to higher performing variants. A critical example is the CAN bus that is still widely used and in many cases already reaching bandwidth limitations. This fact makes it difficult to support the additional overhead of cryptographic signatures in some cases.

The industry is following the tradeoff listed below:

1. Classify CAN messages with respect to security criticality and only apply secure communication for critical messages thereby saving bandwidth.
2. Use more, parallel CAN busses to lower traffic thus gain bandwidth
3. Isolate less security critical domains from domains that are security and safety critical.

Isolation can be achieved by domain specific gateways hosting firewalls principally similar to the aforementioned ECU intrinsic firewall.

Crypto-Agility

Cars have a substantial longer lifetime than typical IT equipment. The time the architecture and the security foundation of a car are defined until the car is put out of operation can exceed 20 years and more. This raises the question whether the used crypto algorithms can be regarded as being secure during such long life time. Institutions like the BSI [15], [16] and others publish estimates on the lifetime of known crypto algorithms and related key lengths. There is an evident risk that the car requires an update of its cryptographic algorithms over the life time. So the question of crypto agility raises up.

Crypto agility has many aspects:

1. The overall software architecture needs to support a structure that enables an easy exchange of cryptographic functionality.
2. The architecture should also prepare future migration strategies e.g. it should simultaneously support newer and older algorithms. Thereby an evolutionary introduction of new algorithms in the car and infrastructure is possible.

3. Busses and storage need to have enough headroom with respect of bandwidth, size and performance to support longer keys.
4. Finally the cryptographic hardware accelerators need to support the new crypto algorithms ideally without impact on real time behavior.

With respect to first and second point mentioned before, the TPM 2.0 standardization already supports crypto agility in all the required specifications. The TPM specification provides a framework which allows adding further crypto algorithms. It is expected that future versions of TPM will therefore support new crypto algorithms while maintaining support for older still relevant algorithms. An ideal solution would be to update specific hardware components step by step, which support new cryptographic algorithms while the backward-compatibility to older algorithms is preserved. This would enable smooth transitions and provides more flexibility on the development process of ECUs, vehicles and supporting security infrastructure.

Automotive Security Use Cases and their Difficulties

In the following, a variety of practical automotive use cases for security will be described and also possible difficulties will be shown that need to be solved or, at least, developers need to be aware of.

They all address the area of typical automotive control ECUs used for gateway, powertrain, body, chassis or occupant safety applications. Infotainment systems, due to their different system architecture, might partially need to implement alternative solutions if they are not in place already.

Security vs. Functional Safety

Automotive Security in itself is a complex topic but it gets even more complicated when functional safety enters the equation.

Unfortunately, a lot of car functions and systems are more or less safety related. Actually when talking about powertrain, chassis or occupant safety systems the majority of ECUs must be safe.

Formerly most of the related functions have been strictly developed according to safety requirements and standards like the ISO26262 [17]. But nowadays they also have to be considered to be security related as the connectivity makes them subjects to attacks.

One key point in the relation between security and safety is that there is no direct mapping from one to the other. This means there is no easy way to derive a required level of security from an already known safety integrity level of a function or vice versa.

Another key point is that the reactions derived from a safety analysis on one hand and from a security analysis on the other hand may be in conflict. While a security mitigation response might suggest shutting off the function, at the same time a safety goal might require keeping a system available as long as possible to prevent critical vehicle behavior. Unlike in classical IT which has the security priorities CIA (confidentiality, integrity and availability) the car requires AIC giving the availability first priority in many cases. For example a powertrain ECU might be expected to shut off the engine when detecting a manipulation indicating a potential vehicle theft whereas this is not carried out while the car is moving as it could impact safety dramatically.

In addition, also different requirements regarding the development process might result in a security mechanism not being considered to be suitable to be part of a safety related part of a system, when its development e.g. does not meet requirements of ISO 26262.

The lesson to be learnt here is that security and safety requirements and system behavior need to be carefully aligned to fulfill the expectations of both sides.

Development and Maintenance

In the past, development started with a new ECU project and ended ideally with start of production or (typically) sometime later after all remaining functional issues have been solved or specifics of later starting vehicle derivatives have been implemented. This might be followed by a short planned maintenance phase for the involved software before the originally responsible development team is disbanded and commanded to new tasks.

This practice needs to be reconsidered when taking security into account. With the IT evolving and a continually increasing amount of computing power available, security mechanisms are aging and new security gaps may arise which weren't predictable during development. Security updates for the software of the affected ECU might be required then. But who will provide them?

Today long term field maintenance of software comparable to the IT domain is not common in the automotive industry. Therefore OEMs and suppliers need to find new models of cooperation to be able to have the required resources available whenever needed.

Software-Over-The-Air Updates

If a solution for the mentioned development and maintenance issue is found, the question arises how to bring the new software updates to the customer. With the increasing connectivity of cars, consumer electronics like personal computers, smart TVs or smart phones can serve as an example for maintenance services. For consumer electronics it is already common to distribute updates over the internet. In the automotive industry this process is called Software-Over-The-Air updates (SOTA).

But this idea has a systematic error. The system architecture of consumer electronics is typically different from the majority of automotive electronics with their deterministic implementation, realtime requirements and restricted resources.

Automotive ECUs are typically developed for a quite static scope of functionality and their resources are adjusted accordingly. This results in flash memories already mostly filled with program code and random access memories being only about the tenth in size of program memories.

In practice most automotive ECUs receive and program new program code in pieces while buffering in RAM with no options to backup the replaced program code as a fallback. Also verifying the validity and authenticity of the new program software often is only possible after completion of the programming process. In addition partial reprogramming is not common either. This procedure carries the risk that it could result in an immobile car when failing on neuralgic ECUs. A typical error would be that an ECU keeps staying in reprogramming mode without serving its main function anymore and, in addition, is incapable of resolving the situation by itself. This results in a situation which would typically require a garage shop to fix the problem.

As of today the existing update procedures have been designed for development, production and workshop purposes. Here the environment is defined and trained staff is involved.

For the future it is necessary to find more robust approaches which potentially bypass the before mentioned resource restrictions. Examples could be expanding ECU resources or, more likely, inventing a centralized update manager inside the car which has enough resources to buffer or store full program updates and verify them before handing them over to the final target for programming [1], [4], [5], [6].

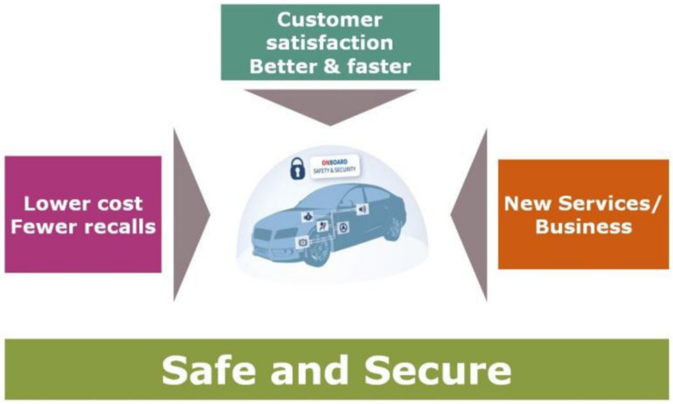

Customer satisfaction Better & faster
ONBOARD SAFETY & SECURITY
Lower cost Fewer recalls
New Services/ Business
Safe and Secure

Figure 10. Benefits of SOTA

The implementation of a well working solution is very desirable as SOTA brings several advantages as it (Figure 10):

- reduces the time to react on bugs and security threats
- is enabling the rapid rollout of new services
- is reducing garage stops thereby saving cost
- ultimately also increases customer satisfaction

Flash Data Security

SOTA is only one out of a bunch of good reasons for another security use case which at the same time could be considered as a security mechanism. Securing data in flash memories of an ECU means ensuring that data to be programmed is supplied by an authorized source and it has not been tampered with. This is both valid for program data and parameter sets stored into flash or other none volatile memory.

Authentication by signatures combined with encryption can help protecting vehicles against the following scenarios:

- Vehicle theft
- Illegal or warranty voiding tuning
- Safety critical manipulation of vehicle behavior
- Unauthorized enablement of fee-based functionality
- Assaults on privacy

For effectiveness of such mechanisms the use of a hardware based trust anchor is suggested.

IP Protection for Apps & Function on Demand

The trend to offer the possibility to permanently or temporarily extend or adapt a vehicle's functionality to its owner's or user's needs or expectations builds another use case for security mechanisms. As the providers of Apps, functionalities or services want to protect their business models from unauthorized copies or unlicensed service usage, there need to be effective countermeasures to prevent such scenarios.

In the past, such adaptions were only done occasionally and mostly also required a change of hardware. The mechanism behind was implemented by simple flags or bit patterns, set in an ECUs nonvolatile memory. Those could easily be manipulated by some diagnostic commands which might have been only protected by a simple global pin, valid for all units of the same type of ECU or even a fleet of cars.

The new trend for only changing or adding software or temporarily enabling pre-defined functions asks for more sophisticated solutions. Those should deliver better protection of Intellectual Property (IP) and business cases and simultaneously be applicable dynamically and remotely in the field without the need for a stop at a service garage.

Using signatures and remote authentication via an OEM back-end, combined with an encrypted storage of configuration data inside the ECU can be an answer to this request. Thereby a secure answer can be obtained if the activation of a function is authorized by e.g. payment of the required fee. In addition the activation or configuration data is securely stored inside the control unit, complicating manipulation attempts. Using a hardware based trust anchor inside the ECU here is mandatory to make the implementation effective.

Also this paper is intentionally not focusing on security mechanisms outside an ECU or the vehicle, this use case just already points out the growing importance of back-end communication for security means.

Centralized Key Management

After description of numerous use cases making use of cryptographic material, also the challenge of handling and managing that material inside electronic control units needs to be addressed. Of course, it would be potentially possible to implement separate mechanisms for handling keys for each different use case. But a centralized management is a leaner approach reducing the handling of key as well as the overall complexity of implementing security mechanisms all over an ECU.

Implementing a centralized key management enables

- A unified API for handling and usage of cryptographic keys
- Generation of separate key material for different purposes which also simplifies later functional extension by providing new key material on demand
- Revocation of local key material dedicated to a specific use case

It allows the combination with ECU specific keys as well as vehicle specific keys as an anchor for deriving further local sub keys on an ECU.

Diagnostics Communication and OBD Interface

Diagnostics Communication over the OBD port exists for almost 30 years and has been primarily used for originally intended service and repair purpose only. Unfortunately in recent years this has changed significantly. The availability of low-price OBD scanners along with the market success of smart phones drives this change combined with the improved availability of small and affordable cellular devices. This development brought a wide range of gadgets fitting to that port and providing access to the car electronics via a choice of wireless connections like Bluetooth, WiFi or even mobile radio. These devices are not only meant for reading fault codes, but e.g. also for

- keeping log books,
- Vehicle locating,
- monitoring driving behavior for insurance purposes,
- efficiency monitoring,
- extension of instrument clusters or
- altering data like resetting error states and inspection warning.

Unfortunately studies [18] have already shown that a lot of those gadgets are poorly implemented including weak or no security mechanisms. But thereby they are an ideal entrance gate for potential attackers by providing easy remote access to the cars network. This opens a door for all kinds of illegal manipulations which in extreme cases also could affect safety negatively.

The diagnostics communication provides access to a lot of functionality originally meant for repair, adjustments, car production or development for authorized and skilled staff in a defined environment. A lot of damage can be caused if used by wrong hands, especially while the car is on the road.

Therefore it is suggested to

- restrict access over the OBD port to the specified OBD functionality only (no extended manufacturer diagnostics)
- keep direct connections to internal busses away from the OBD port
- carefully implement preconditions for activation of diagnostic functions
- omit implementation of potentially harmful diagnostic functions wherever alternatives are possible
- protect essential but potentially risky diagnostic functions against unauthorized access

Possible countermeasures could be:

- Using a gateway architecture for the bus network or a firewall to connect the OBD port to the network
- Implementation of different individually protected diagnostic modes for different ranges of functionality using cryptography based key mechanisms for security

Handling of Field Returns

Analysis of defect parts returned from the field is a common task for the automotive industry. For the car electronics this typically requires a communication port to be able to interact with the ECUs.

Depending on the individual defect and the required level of analysis a choice of the following needs to be supported:

- Workshop diagnostics protocols (with production and developer modes included)
- Extended network messages used during car development
- ECU manufacturer specific protocols or interfaces also used during ECU production
- Access to debug interfaces used for software development

Unfortunately all of these options could potentially be misused by attackers to manipulate the ECU behind as they typically offer a wide spread and rarely limited access to the ECU's resources due to their originally intended usage.

To prevent security breaches these communication interfaces need to be completely deactivated or, in case of workshop diagnostics, restricted to absolutely necessary capabilities. But to maintain the possibility to analyze field returns secure reactivation needs to be possible ensuring that only authorized personnel gets extended access to the ECUs. Furthermore, in this use case trust anchors can help to increase the level of security. They can support the authentication of reactivation requests, which allow to reactivate the full functionality of communication interfaces for authorized personnel.

Summary/Conclusions

Automotive security is prerequisite to enable both already existing and new use cases in the automotive industry. Safety, the high priority of availability and demanding real-time behavior require new approaches when moving IT security concepts to the automotive industry. Nevertheless existing security standards should be reused as far as they are compliant with automotive requirements.

Proven approaches like hardware trust anchors can help to improve security and make its implementations more efficient.

Additionally new measures are required to implement security in automotive.

Likewise the automotive industry has to prepare providing security services for sustaining security for the expected life time of vehicles. Although this paper described various use cases and working points regarding Automotive Security, in conclusion it needs to be emphasized that security can't be solved at a single point, on single ECUs or inside vehicle. Instead it needs a holistic view also involving development, logistics, production and maintenance processes as well as the IT infrastructure inside the involved companies to fully address it. Therefore it is also a challenge across several working fields.

References

1. IHS Inc., "Over-the-air Software Updates to Create Boon for Automotive Market", http://press.ihs.com, Sep. 2015.
2. Escherich, R., Ledendecker, I., Schmal, C., Kuhls, B. , "SHE -Secure Hardware Extension – Functional Specification", Version 1.1, Hersteller Initiative Software (HIS) AK Security, Oct. 16, 2009.
3. Weyl, B.; Wolf, M.; Zweers, F.; Gendrullis, T. ; "Secure on-board architecture specification", EVITA Deliverable D3.2, Aug. 2011.
4. Lobdell, M., "Robust Over-the-Air Firmware Updates Using Program FLASH Memory Swap on Kinetis Microcontrollers", Freescale Application Note, AN4533, Rev. 0, Jun., 2012.
5. Steurich, B., Scheibert, K., Freiwald, A., and Klimke, M., "Secure and seamless integration of Software Over The Air (SOTA) update in modern car board net architectures" SAE Technical Paper 13th ESCAR Europe, 2015.
6. Steurich, B., Scheibert, K., Freiwald, A., and Klimke, M., "Feasibility Study for a Secure and Seamless Integration of Over the Air Software Update Capability in an Advanced Board Net Architecture," SAE Technical Paper 2016-01-0056, 2016, doi:10.4271/2016-01-0056.
7. Heartbleed bug: https://en.wikipedia.org/wiki/Heartbleed
8. Trust Computing and Heartbleed: http://www.trustedcomputinggroup.org/avoiding-heartbleed/
9. Financial industry affected by Heartbleed: http://www.fsroundtable.org/financial-services-industry-swats-heartbleed-bug/
10. Common Criteria Main page: https://www.commoncriteriaportal.org/
11. Lomne, V., "Common Criteria Certifications of a Smartcard: a Technical Overview", CHES 2016, Santa Barbara, USA, 2016, http://www.chesworkshop.org/ches2016/presentations/CHES16-Tutorial1.pdf
12. Arthur, W., Challener, D. and Goldmann, K. "A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security", Apress, 1st Ed., 2015
13. Tillich, S., Wojcik, M., "Security Analysis of an Open Car Immobilizer Protocol Stack",Springer 2012, https://eprint.iacr.org/2012/617.pdf
14. Indesteege, S., Keller, N., Dunkelmann, O., Biham, E., Preneel, B., "A Practical Attack on KeeLoq", Advances in Cryptology – EUROCRYPT 2008, Springer, 2008
15. Giry, D., BlueKrypt - v 29.2 -, 2015, https://www.keylength.com/en/8/
16. Bundesamt für Sicherheit in der Informationstechnik (BSI)-Technische Richtlinie, "Kryptografische Verfahren Empfehlungen und Schlüssellängen (BSI TR-02102-1)", 2016, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?_blob=publicationFile
17. "ISO 26262:2011: Road vehicles - Functional safety" Part 1 to 10, International Standardization Organization, November 2011, http://www.iso.org
18. Klinedinst, D., King, C., "On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle", Carnegie Mellon University SEI Digital Library, March 2016, http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=453871

Contact Information

Rolf Michael Schneider
AUDI AG
85045 Ingolstadt
Germany
rolf.schneider@audi.de

Martin Klimke
Infineon Technologies AG
85579 Neubiberg
Germany
martin.klimke@infineon.com

Udo Dannebaum
Infineon Technologies AG
85579 Neubiberg
Germany
udo.dannebaum@infineon.com

André Kohn
AUDI AG
85045 Ingolstadt
Germany
andre.kohn@audi.de

Acknowledgments

Many thanks to Karin Schmidt from Infineon for being our English trainer.

Definitions/Abbreviations

AES - Advanced Encryption Standard

CAN - Controller Area Network

CRC - Cyclic Redundancy Check

ECC - Elliptic Curve Cryptography

ECU - Electronic Control Unit

FBL - FLASH Boot Loader

HSM - Hardware Security Module

MCU - Micro Control Unit

OBD - On Board Diagnosis

OEM - Original Equipment Manufacturer

OTA - Over The Air

RSA - Rivest-Shamir-Adleman cryptosystem

SFBL - Secure FLASH Boot Loader

SHA - Secure Hash Algorithm

SHE - Secure Hardware Extension

SOTA - Software Over The Air

SPI - Serial Peripheral Interface

TLS - Transport Layer Security

TPM - Trusted Platform Module

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. The process requires a minimum of three (3) reviews by industry experts.

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE International. The author is solely responsible for the content of the paper.

ISSN 0148-7191

http://papers.sae.org/2017-01-1652