Cryptographic Component Identification: Enabler for Secure Vehicles

André Weimerskirch, Christof Paar and Marko Wolf
escrypt – Embedded Security GmbH
D-44801 Bochum, Germany
{aweimerskirch, cpaar, mwolf}@escrypt.com

Modern vehicles embed several dozens of electronic control units, infotainment devices, and safety relevant components. All these devices might be manipulated, counterfeited, stolen, and illegally exchanged. Hence, a proper identification of components would be valuable. In this work we propose protocols for component identification in a vehicle. The component identification provides protection against manipulation, counterfeits, and theft by implementing a mechanism to bind security relevant components to a given system. Additionally, such component identification enables secure inter-vehicular networks as well as innovative technologies such as an electronic license plate.

Keywords: component identification, cryptography, trusted computing, TPM

I. INTRODUCTION

Component identification enables a vehicle to check whether all built-in components are proper. Thus, it directly provides protection against manipulation, counterfeits, and theft. Furthermore, component identification might be used as basis of innovative technologies. Several scenarios and protocols regarding security of smart vehicles were introduced by Hubaux et al. [4]. A basic enabler for such scenarios is a unique identifier for each vehicle provided by an electronic license plate. A core requirement is that the hardware module that implements the electronic license plate cannot be manipulated nor faked or stolen in order to install it in another car. The same requirements hold for all security related components. Clearly, component identification provides a mechanism to bind such components to a given vehicle. Now, if an adversary manipulates or removes any component of the vehicle, the vehicle is able to notice that fact and therefore able to react.

To the author's knowledge there is no system yet that provides component identification using cryptographic methods. Traditional methods use tags, e.g. holographic stickers that are supposed to be unforgeable. However, as it can be seen in the airplane industry, such tags can easily be bought on the black market [2]. Another way of providing a basic protection against counterfeits and theft is ensured by mechanical countermeasures. For instance, there are car radios of special size such that they only fit into cars of a single manufacturer. However, that solution is uncomfortably.

In the following we propose a scheme for providing component identification in order to bind security and safety related parts as well as special components of a vehicle to a central security module that is crucial for the vehicle. New components can be added and replaced, and components are provided with a secure communication channel among themselves. We use component identification to secure all components of a car against cloning (faked parts), manipulation, and theft. In particular, our scheme protects the car owner of bogus parts that endanger the operation of the car, and it minimizes the financial damage of suppliers. The worldwide damage of counterfeits is estimated to be as high as 3 trillion US Dollar worldwide [5]. Our protocol is based on a central security module which we call a hardware security module (HSM). An HSM is a tamper resistant device based on a smart card microcontroller [6] or a trusted platform module (TPM) [10] that is able to perform cryptographic operations securely. For instance, such an HSM might be an enhanced and secured engine management unit. In the full version of this paper we present also solutions where no central HSM is required but where the role of the HSM is distributed to all components instead.

We believe that our scheme can be implemented in a very cost-efficient manner by attaching an RFID transponder to components. Our solution is universal and easy to apply. There is no need to adjust components before they are installed, and all processes can be executed automatically. Furthermore, our scheme provides a secure framework for all involved parties including the car owner as well as the manufacturer, suppliers, and service mechanics. Our solution is applicable to cars but also to other transportation vehicles such as trains and airplanes.

A. Assumptions

For the remaining we assume the following.

- A simple computing tag is attached to each crucial component in such a way that removing the tag destroys the component. In the full paper we show how a passive RFID transponder can be used as such a tag by using symmetric cryptography only. RFID tags supporting AES are already available today [3].
- The HSM as well as the computing tags are able to perform cryptographic operations. The HSM is a crucial component of the car. Without the presence of the HSM component, the car will not run.

- There is a temporary connection available between the vehicle and a central server. This assumption can be omitted at the cost that protection against cloned parts is not possible then.
- The component tags communicate over a wireless communication channel. All components are able to communicate to each other component and to the HSM regardless whether a direct single-hop or multi-hop communication channel is required.
- To perform cryptographic operations and to transmit data packets the component tags need quite some energy. Hence, we assume that they are equipped with a battery having a life span of several years that outlasts the lifetime of the component, or that the tag is directly powered by the component.
- There is a trusted third authority that issues certificates, e.g., a government institution or a car manufacturer.
- The components or the central security module are able to take actions in case that a protocol step fails. For instance, the central security module might share a key with crucial car components. If any check in the protocol fails, an alarm is issued.
- Each component holds a unique certificate <*PK, ID*> consisting of the identity *ID* as well as a cryptographic public key *PK*. *ID* comprises a unique identification string as well as further information such as the manufacturing date and the component's quality class (see [1] for more details about asymmetric cryptography and public key infrastructures). The component furthermore knows the secret key *SK*.
- The HSM of a vehicle holds a list *UL* of all components built into that vehicle. *UL* is regularly synchronized with a global list *GL* of all components. The synchronization is performed in a secure manner to avoid any manipulation. Each component can be uniquely identified.
- There might be a global certificate revocation list *CRL* with all components that were revoked. A component might be revoked if it was stolen, or if it is known that it was cloned.
- The cryptographic mechanisms are properly implemented such that attacks on the implementation are not possible. For instance, so called side-channel attacks that are based on the particular running behavior in order to obtain the secret key are not possible.

Clearly, the scheme becomes especially powerful if all components have a TPM chip embedded.

II. Component Identification

The main goal is to bind components securely to a vehicle. We now present a solution based on an HSM using asymmetric cryptography. The main threat for a car arises if components are stolen, manipulated, or cloned. We define *cloning* as the act of rebuilding a component up to a perfect copy, in some cases even including secret cryptographic keys. Hence, our scheme provides piracy protection, system protection, and theft protection. *Piracy protection* provides the ability to identify original parts and the possibility to detect counterfeits and cloned components. *System protection* provides system integrity by monitoring the system for unauthorized changes, and *theft protection* prevents the use of stolen components in another system.

We are considering a system in which all components hold authentic data. There are three phases that we take into account: (1) The installation of a component into a car, (2) the running system, and (3) the demounting of a component out of the car. We describe our security goals by an example. Only original parts can be built into the car. Every time the ignition key is turned the system integrity is checked, and only in case of an unaltered system the engine starts. A display in the dash board shows the status of all present system components. This could be useful, e.g., to prove that the mechanics of a car garage used original parts only. There might be also a button to manually start the check such that the owner can prove integrity to a third party. For instance, the owner can prove to police that a proper electronic license plate is built in.

All identifications are performed in a challenge-response manner. After *A* presents a certificate, *B* needs to check whether *A* has knowledge of the corresponding secret key *SK*. A standard challenge-response method using digital signatures can be found in [7]. The check of a symmetric key *K* is performed in a similar way by using a message authentication code (MAC). A standard challenge-response method for providing this check in a mutual way can also be found in [7]. Here, *A* checks whether *B* knows the secret key *K* and also *B* checks whether *A* knows the secret key. Note that in general all messages of components are encrypted and authenticated by a commonly shared key *K*.

As already mentioned there are three phases to consider when talking about component identification which we will discuss in detail. The main idea is to have an initial component, e.g., the HSM that is imprinted with a secret key *K*. Each component holds a certificate. If a component is added to the vehicle, the certificate is checked. Then the component obtains the secret vehicle key *K* which is checked while the car is running in order to prevent manipulations after the installation. Finally, our solution allows a controlled demounting of a component in order to distinguish stolen components from properly demounted ones. The life cycle of a component is depicted in Figure 1.

A. Initialization

At initialization time, the HSM is installed into the car as first component. Further components might be installed into the car at initialization time. If the HSM requires means of disabling the car or issuing an alarm the HSM exchanges secret keys with crucial components. For instance, the HSM might agree on a key with the car's engine and the car's dashboard, e.g. by using the Diffie-Hellman key agreement [7] based on the HSM's and components' certificates such that it is able to disable the car or to display warning messages. Note that for this purpose a separate key is shared between the HSM and the component that is only used for this purpose.

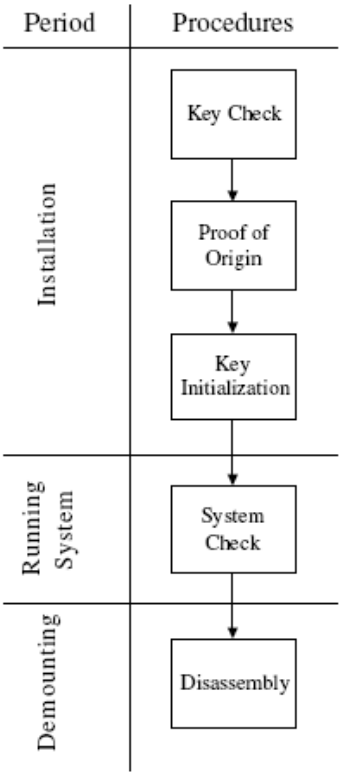

Figure 1: Life cycle of a component

The HSM now randomly chooses a key *K* that becomes the vehicle key. Assuming that the car is assembled in a trustworthy environment such as a manufacturer plant the vehicle key *K* is now distributed to all installed components. Note that here the key *K* is not encrypted. If there is no trustworthy environment available then for each installed component the installation procedure can be executed as described below.

If the component tags are equipped with computationally strong CPUs a traditional key management for the formation of ad-hoc networks can be used. For instance, a group key-agreement scheme based on the components' certificates could be used for a higher security level at higher computational costs [9]. For the devices we envision here we believe such a scheme to be too resource demanding, though.

B. Installation of a Component

A new component is installed by adding it to an already existing set of components that form the car. Components are added stepwise. Once a new component is added to the car the installation phase is executed. It consists of the following steps:

1. *key check:* check whether the component is already part of the vehicle, e.g., after the component was disassembled for repairing it.
2. *proof of origin:* check whether the component has a valid certificate.
3. *key initialization:* providing a valid component with the vehicle key *K*.

The key check runs as presented in Figure 2. First, the newly installed component *C* provides its unique *ID*. The HSM checks whether $ID \in UL$. This holds if *C* was part of this car before. If so, the HSM checks if the new component knows *K* by a challenge-response authentication. Otherwise, if the component is new, the proof of origin check is performed.

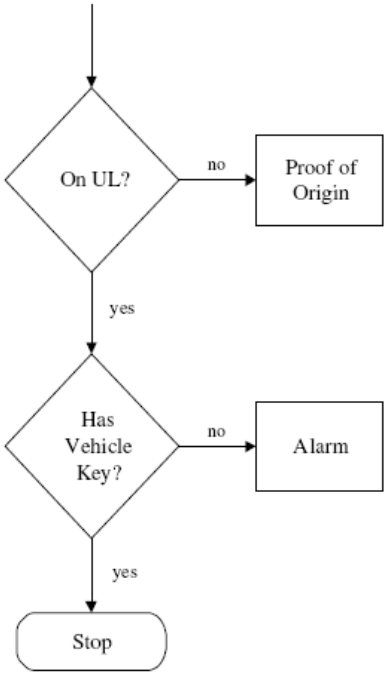

Figure 2: Key Check

During the proof of origin, the HSM checks whether the new component provides a valid certificate. During the proof, the new component *C* is put on *UL* in a preliminary manner. If C is on the global built-in list *GL* then it is already built into another vehicle which indicates that this component was cloned or stolen.

Finally, after all checks were performed successfully, *C* is provided with the vehicle key *K* during the key initialization. For that reason, the HSM sends *E(K, PK)*, which is the encryption of *K* by the key *PK*, to *C*.

Clearly, only components that have a valid certificate are able to obtain the vehicle key *K*. It follows that all components that know *K* are trustworthy and play fair, i.e., they do not compromise the system later on.

C. The Running System

We ensured that all components that are installed into the car were verified. As trustworthy components they were initialized with the vehicle key *K*. However, after they are installed they might be manipulated, exchanged, or removed. Thus, we execute the system check in order to verify whether all components know the vehicle key *K*. The system check can be executed several times:

- every time the car is started
- periodically, e.g., every 30 minutes

- initiated manually, e.g., to prove system integrity to a policeman

There are basically two methods to execute the system check. In the first version, which is depicted in Figure 3, the HSM challenges all components one after the other by a challenge-response method to check the components' knowledge of *K*. Clearly, only components that are listed in *UL* are involved.

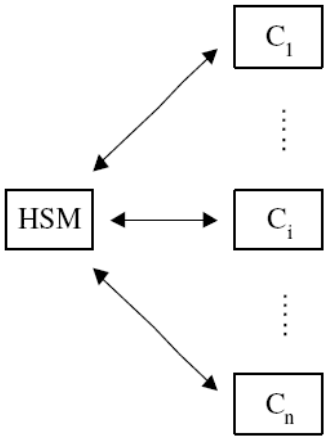

Figure 3: System check with one verifier

It is also conceivable that each component of the vehicle checks another component. The HSM starts by challenging another component, which in turn challenges another component, and so on, until the security module is challenged by a component such that the circle is closed. Here, each component holds the list *UL* of all installed components, and all components have to agree on an order of being challenged. If any check fails, the challenging component raises an alarm by sending a message to the HSM. The HSM then takes appropriate actions. Note that if the communication bus supports parallel communication, the system check can be performed in parallel by all components. Then all components send their challenge and respond to a challenge in parallel. Obviously, it is easy to prevent an attacker from jamming the communication channel in order to delete the alarm message by letting each component either send a positive or negative alarm message. By using a challenge-response method, replay attacks are prevented.

Clearly the system checks are vulnerable to compromised components. A component might always fake an authentication since all components use the same shared key *K*. Furthermore, in the cyclic version a component might "short-cut" an authentication process by skipping components within the mutual authentication cycle. However, we concluded that all components that have knowledge of the system key *K* play fair. If *K* is known the system can be considered to be broken. The check can also be performed based on the component's certificates. Such a check is not vulnerable to an insider attack where the insider has knowledge of the system key *K*. However, such a check based on asymmetric cryptography is in many cases too demanding for the considered device class.

D. Demounting of a Component

The demounting procedure has to be performed once a component is removed from a vehicle in order to install it into another vehicle. If a component is removed and then installed again into the same car, nothing needs to be done. Hence, the demounting procedure is performed to be able to distinguish a legal demounting to a theft of a component.

To remove a component *C* in a controlled way, it first has to prove that it knows the secret vehicle key *K*. Then *C* is deleted of the system and of the built-in lists *UL* and *GL* such that it appears like a new component. Components which are bound to a specific car and thus must not be demounted can be marked as such in their certificate, the *UL*, and the *GL* by a flag. The HSM will then not allow these components to be demounted from the vehicle.

In cases where there is no *GL*, a component stores a flag in its memory which indicates whether the component is installed in a car. When the component is demounted, the flag is deleted. Components can only be installed into a car if the flag is not set. Such a solution is also preferable in cases where the synchronization of *UL* and *GL* is performed only rarely.

III. EXTENSIONS AND FEATURES

In the previous section, we presented the basic component identification mechanism. There are several extensions possible to improve the methods considerably. These are explained in all details in the full version of this paper.

As explained before, a central secure hardware module, namely the HSM, is required. The HSM is a central point of failure and attack and also quite expensive. However, it is possible to *distribute* the role of the HSM to all the components. Then all components perform the verification of newly assembled components and of raising an alarm in case of an exception. Hence, any component would be connected to any other component such that breaking the system would require replacing all components. Such a scheme reduces cost considerably since an HSM is not necessary anymore.

A further extension is to use *RFID tags* as the above described component tags. Passive RFID tags do not require an external power source but are extremely resource constrained. However, we believe that it is possible to use RFID tags embedded into the components in order to perform the required cryptographic operations [8]. We modified the protocol suite in such a way that it only requires symmetric cryptographic operations. Such operations can be performed very efficiently since they are by some order of magnitudes faster than asymmetric cryptographic operations. Such RFID tags are very cheap and they have a large lifespan.

Our component identification system can be used in a highly flexible way by introducing *policies*. A component's certificate might contain its role for the system such that all components obtain this information at the point when the new component is installed. If there is a failure later on regarding this component, the remaining components then might act in a way described by their policies. For instance, if the airbag of a car is missing, all the other components might not work since a security related part is missing. However, if the car radio is missing,

only the car's head-unit might display a warning message to the driver. Our system is flexible in several ways. For instance, it might be used to track the *component's history*. For each component it would be possible to exactly track down when it was produced, where it was built-in, and where it was repaired. Such a component history would especially be useful for systems with very expensive spare parts such as airplanes.

Our scheme also allows *access control* when repairing or replacing spare parts. For instance, consider an airplane. Each mechanic needs to insert a personal smart card to the airplane before replacing spare parts. However, each mechanic might only have access to some spare parts but he might not be allowed to replace any component. Furthermore, it might be necessary that another mechanic always agrees in replacing a given component such that always two mechanics need to insert their smart cards before a component can be replaced properly. The head-unit of the airplane would then protocol the history of all components. If any component is replaced without authorization an alarm is raised. Furthermore, *key hierarchies* might be introduced. Consider a vehicle. The owner might have a smart card that allows him to replace non-security related parts. However, only the garage mechanics have another smart card allowing them to replace security related spare parts such as breaks. A nice side effect is that the vehicle will protocol all activities such that a garage cannot charge for spare parts they had not replaced.

The scheme is based on standard cryptographic mechanisms. It would be possible to bypass the scheme by replacing all components. Replacing only a single component, even the HSM, does not suffice since all remaining components will then take actions. Hence, replacing all involved components needs to be more expensive than the potential gain.

IV. Conclusions

We presented a scheme for implementing component identification in vehicles based on cryptographic authentication schemes. Our solution is mainly based on the tamper resistance of special security hardware modules such as smart card microcontrollers or a TPM chip. In cases where the component has a mechanic main function, e.g. an engine, it has to be researched how secure hardware modules can be embedded into a component. If the component has mainly an electronic function such as a dashboard instrument, the security tag which is in the best case a TPM can be embedded into the component. In such cases, trusted computing mechanisms are able to provide tamper resistance. Since more and more parts of a car depend on electronics, using such a solution will provide the required means for our component identification scheme to easily protect the entire vehicle.

References

[1] C. Adams and S. Lloyd, "Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations", New Riders Publishing, 1999.

[2] Business Week, W. Stern, “Warning! Bogus parts have turned up commercial jets. Where's the FAA?", 1996.

[3] M. Feldhofer, S. Dominikus, J. Wolkerstorfer; "Strong Authentication for RFID Systems using the AES Algorithm", In Proceedings of Workshop of Cryptographic Hardware and Embedded Systems - CHES 2004, Volume 3156 of Lecture Notes in Computer Science, Springer, pp. 357–370, Boston, USA, August 11-13, 2004.

[4] J.-P. Hubaux, S. Capkun and J.Luo, "Security and Privacy of Smart Vehicles", to appear in IEEE Security & Privacy, 2004.

[5] Gieschen Consultancy, "Report: IP Theft up 22%, massive $3 Trillion Counterfeits", May 2005.

[6] O. Kömmerling, M Kuhn, “Design Principles for Tamper Resistant Smartcard Processors“, In USENIX Workshop on Smartcard Technology proceedings, Chicago,USA May 10-11, 1999

[7] A. Menezes, P.C. van Oorschot, S. A. Vanstone, „Handbook of Applied Cryptography“, CRC Press, 1997.

[8] S. E. Sarma, S. A. Weis, D. W. Engels: RFID Systems and Security and Privacy Implications, In Proceedings of Workshop of Cryptographic Hardware and Embedded Systems - CHES 2002, Vol. 2523 of Lecture Notes in Computer Science, Springer, pp. 454 - 469, Redwood Shores, USA, August 13-15, 2002.

[9] M. Steiner, G. Tsudik and M. Waidner, “CLIQUES: A New Approach to Group Key Agreement", In Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS'98), 1998.

[10] Trusted Computing Group (TCG), “TPM Main Specifications Version 1.2”, www.trustedcomputinggroup.org, June 2005