A Secure but still Safe and Low Cost Automotive Communication Technique

Rafael Zalman
Infineon Technologies
81726 Munich
Germany
+49 89 234 86209
rafael.zalman@infineon.com

Albrecht Mayer
Infineon Technologies
81726 Munich
Germany
+49 89 234 83267
albrecht.mayer@infineon.com

ABSTRACT

In this paper, we firstly give an overview of the security perimeter in modern automotive systems and propose then a cost effective solution for authentication of communication data. The proposed solution provides end to end protection, it covers the aspects data content and generation time (freshness) and it can be implemented for different standard communication busses without a bus protocol change. Its low overhead makes it in particular suited for short data messages of real time systems, like messages on bandwidth restricted automotive buses.

Keywords

Safety, security, automotive, on-board communication, authentication, freshness.

1. INTRODUCTION

Modern automotive systems are becoming increasingly complex and connected. Nowadays vehicles contain several ten of electronic control units (ECUs) with powerful processing capabilities, running different SW and HW architectures interconnected via different types of on-board network infrastructure. The traditional design of vehicles was until recently focusing on the standard functionality like engine management, steering and breaking. The next step was the inclusion of safety engineering (which has also in between reached a certain maturity including standard architectures like lockstep processing) aiming to increase the safety of modern vehicles to the latest norm requirements. Developments like anti-lock brake systems, redundant buses and the general constant inclusion of safety related elements in all the automotive systems made cars much safer for their passengers and for other traffic participants. Maturity in the functional safety domain is also marked by the publication of an own automotive functional safety standard (ISO26262) in 2011. However, the entire functional safety domain is not considering the existence of any situation in which the generation of a fault is the result of an intentional attack. The assumed fault model is based on statistical physical defects and of unintended systematic faults induced during the development cycle. Independent from safety considerations some security measures like tuning protection were introduced since the late nineties. Such protection measures shall prevent any system change which is not authorized by the car manufacturer, like engine performance upgrades, or the addition of un-intended features (existing in the SW but not enabled on a particular ECU).

Besides the traditional on-board communication networking (e.g. based on different CAN flavors or FlexRay), more and more vehicles offer also wireless interfaces ranging from infotainment near range connections (e.g. mobile phones via Bluetooth or WiFi) to telematics infrastructure services (like automatic crash response and remote diagnostics via cellular links). This accessibility of the vehicle from the outside creates a full range of new opportunities for a potential attacker offering a full new range of attack surfaces.

Therefore, recently automotive security became an important field of research and received a lot of focus from the OEMs and the entire value chain of vehicle industrial production. The vulnerabilities showed in many studies of commercial available cars (like [2] and [3]) implied the immediate need of new techniques and measures for increasing the security level of the current automotive implementations.

2. ATTACK SURFACES

While the multitude of attack surfaces increases with each new feature or electronic module included in a modern car, some classification can be done based on a logical partitioning of the domain. However, there are many ways in which one can categorize the possible security attacks on a vehicle.

One first security view / division can be defined as the ECU seen as a separate minimum entity on which security techniques can be implemented and maintained. In this view, the security goal of the ECU would be to protect its own SW and data. According to this viewpoint, the purpose is to avoid the alteration of the SW by a malicious one and of course the protection of the security assets (e.g. keys). In general, this type of protection is based on specialized HW (e.g. High Security Modules) available on some microcontroller architectures supported by the corresponding SW architectures.

Another view relies on the in-vehicle network perspective in which a vehicle is seen as a network of connected ECUs. These ECUs are providing the intended functionality and safety requirements for the vehicle and they communicate with each other via signals transported on the different bus systems available in the car.

DAC '14, June 01 - 05 2014, San Francisco, CA, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-2730-5/14/06…$15.00.
http://dx.doi.org/10.1145/2593069.2603850

In this view, the security goal is to protect the integrity of the signals transmitted between ECUs on this in-vehicle network against un-authorized and potentially malicious manipulation. Special care shall be allocated to the safety-relevant signals transported on these internal buses which, if maliciously tampered can cause the violation of a safety goal for the entire vehicle (e.g. braking or suspension control).

In general, in this ECU network view, the communication takes place between different ECUs or ECUs and sensors connected also on the buses – from the communication itself, there is no difference in the signal transport. Special cases in which the access to the bus is granted via diagnostic interfaces (e.g. OBD-II) have to be considered as these interfaces are mandatory in some areas (e.g. in US this interface is federally mandated).

A solution allowing the protection of the signal integrity, authenticity and real time properties in an automotive network of ECUs, is the subject of this paper.

Finally, the last view of a vehicle may be "a vehicle as a node in a wider network". This view considers the whole vehicle as a connection point (or more connection points) in a network consisting of either other vehicles (like in the case of car-to-car communication) or standard infrastructure (e.g. internet connections to content servers, services, etc.). These connections are enabled either via vehicle build-in wireless communication channels, or via user mobile phones serving as gateways, or both.

In this case, of a car seen as a node in a wider network, the security goal would be clearly to protect the safety-related properties and integrity of the vehicle against malicious attacks. One more requirement in this case consists in the protection of the driver / vehicle occupant's privacy. Recent studies [2] showed that it is possible to actively interact with the vehicle's brakes, instruments, or engine control completely remote which poses a very high risk on the safety of the vehicle. These kinds of attacks via internet represent a real threat with their potential to affect a large number of similar vehicles.

3. SAFE AND SECURE COMMUNICATION

As mentioned already, this paper focuses on the in-vehicle network (or on-board communication) in which the car is seen as a network of ECUs.

3.1 Examples

One very vulnerable point of a car seen as a network of ECUs is the aggregation of information from more than one ECU for complex functions. A classic example here is the Electronic Stability Control (ESC) which aggregates information from different sources like sensors (e.g. accelerometers) speed (each wheel), throttle angle, steering angle, etc. and also controls some of the vital systems for the vehicle trajectory like traction and steering feedback. Complex interactions are needed also with the Anti-lock Braking System (ABS) in order to maintain a safe trajectory in case of difficult driving situations (skid, roll dangers, etc.). One even more recent development is the Active Cruise Control (ACC) in which the information of the traffic ahead of the vehicle is processed and, depending on the implemented algorithm, the system is using the network to control the throttle / traction and may also control the brakes (some vehicles will automatically brake stopping the vehicle if needed without any driver intervention). Other recent implementations in this direction include the automatic parallel parking in which the car is executing complex maneuvers without driver intervention and in which the network data flow plays a crucial role, or the pre-crash actions in which the propagated information about an imminent impact is used by the networked ECUs for brake preparation, seat-belt pre-tensioning, etc.

In all the mentioned systems, the information flowing on the network is clearly safety-relevant and any malicious tampering with this data can have a very high associated risk. Due mainly to costs and legacy, separate buses transporting only safety-relevant data for this kind of function aggregation is not feasible so common infrastructure shall be used. As of today, the main vehicle infrastructure is still based on different flavors of CAN, LIN and FlexRay buses with Ethernet struggling to enter this application field. For safety related signals exchanged over the vehicle network, several solutions were implemented (e.g. [1]) which, given a certain well defined statistical fault model which may influence the data transported on the buses, will ensure the required data integrity and time consistency. This is done in general by adding redundant information in the transported signals (depending on the assumed statistical fault model different methods are available e.g. CRC codes with different lengths and protection properties) and including a sequential counter scheme in order to ensure the time consistency. The problem with such approaches is that this does not assume any malicious attack on the network but rather a statistical distribution of physical faults which may impact the network signals with a certain probability and according to a certain distribution ([4]) allocated to the physical channel. This approach is of course not suitable in the context of a potential malicious attack as the safety mechanisms are in general reversible – this means that an attacker can reconstruct the correct signal from the redundant information added to the transmitted signal.

3.2 Requirements

One important aspect is that the participants in the network are not necessarily all ECUs (in other words, units with a reasonable computing power or having HW accelerators for mathematical intensive computation patterns) but part of the nodes consist of (intelligent) sensors with rather limited resources available and hence not suited for complex cryptographic algorithms.

From security point of view, the focus is to protect the authenticity and integrity of the signals transmitted on the network between the network nodes. In this context, the authenticity resides in the capability of the receiver to determine that a certain message was sent by a trusted sender by analyzing the message content. The integrity of the message consists of the correctness of the message i.e. the received message is identical with the sent one. In this case, the integrity has a slightly different "flavor" as in the functional safety integrity: the fault model is not assumed as a random fault of the channel with a known statistical distribution but rather the assumption is of an intentional, malicious information corruption. So the requirement here would be that the receiver shall be able to determine if the message is "integer": it has the same content as the one sent by the trusted sender.

In this context, we will associate in the integrity property of a message also the correctness of its time characteristics due to the usual (hard) real-time constraints of the large majority of automotive applications. This means, the receiver shall have the possibility to determine if a message is also correct from the time point of view – if the message arrived in the expected (and assumed correct) time window interval it was supposed to. This property of the message can be also found in the literature as "freshness" of a message. This requirement is mainly derived from the possibility of "Replay Attacks" in which an attacker is re-using recorded correct information which already transited the network in different moments of time which would otherwise seem authentic and integer for the intended receivers. A typical use case for such an attack would be messages containing measured values from sensors. By introducing the time dimension, a message will be considered integer only if it is authentic, passes the integrity check and is also fitting in the expected time window.

Except for the encoding scheme, the rest of the message is transmitted un-encrypted. The solution focuses on authentication and integrity of the messages and not on the confidentiality of it for reasons related mainly to availability (making the system tolerant to potential faults in the security computation intensive areas) and legacy (coexistence of different generation of ECUs in the same network: some may have implemented the authenticity and integrity mechanisms and some may not have this feature available yet).

Examples of such messages which are transmitted on the network may be related to vehicle access (like the signals related to the alarm systems, the opening of doors, trunk, windows, etc.), to car configuration (preventing the un-authorized exchange of ECUs in a vehicle network by using special pairing-codes) or to driver assistance signals (like the ones for ESC, ACC, lane detection, etc.).

3.3 Entry Points for Attackers

For an attacker the access to the ECU network can be:

- Direct physical access to the vehicle network. This may be in practice quite difficult but simpler network access ways like the connection via OBD vehicle interface have to be considered. This can be used for a complete system analysis which enables an attack using the access points, described in the following.
- Malicious component inserted in the vehicle's network (e.g. after-market options like CD players, digital radios, etc.). In this case, an ECU participating to the network communication has a malicious SW which has full access to the entire transferred communication on the network.
- Access via other (remote) interfaces. These attacks are using the wireless interfaces (TPMS, Bluetooth, WiFi, GSM, etc.) equipping any modern as a gateway to the internal network. The attack is using these interfaces to gain control over an ECU connected to the internal network hence offering remote access possibilities similar to the physical access.

3.4 State of the Art Solution

A typical solution is using a pre-shared symmetric key Message Authentication Code (MAC) which, applied to the messages (or on pre-agreed parts of it) transmitted on the network, allows the authentication of the transmitter while ensuring also the integrity of the message. One example of such a MAC implementation is a CMAC ([5]) block cipher implementation based on the "Advanced Encryption Standard - AES" ([6]). This approach has the implicit advantage of allowing a truncation of the transmitted MAC thus offering a good balance between security protection and network traffic.

The encoding scheme for a network transmitted message can be described in principle by:

- Based on the pre-shared key, a MAC is calculated on the message data and optional a message counter value.
- The computed MAC (or a pre-agreed truncated MACt) is added to the message frame.
- The message is now transmitted on the network using he standard transmission protocol (e.g. CAN, FlexRay) as specified for example in [1]. This transmission will add by default the integrity check codes (e.g. CRC) against corruption on the network (statistical corruption, not intentional).
- At the receiver side, the MAC is re-calculated based on the received data and optional on a local message counter value and compared with the transmitted MAC. If the values are identical, the message is authentic and integer with respect to the data content.

4. PROPOSED SOLUTION

The algorithm, described in the last section has three disadvantages.

- The time information is not included, so a message delay attack is feasible.
- For small amounts of data (e.g. a sensor value) MAC and CRC are very significant message length adders
- The calculation of the MAC is in the real time path between sender and receiver, so the crypto HW needs to be fast enough and available if it is shared.

The proposed solution addresses these disadvantages. Figure 1 shows the basic approach. It can be varied for different optimization targets. This will be done for two examples at the end of this section. In the following the basic steps are described.

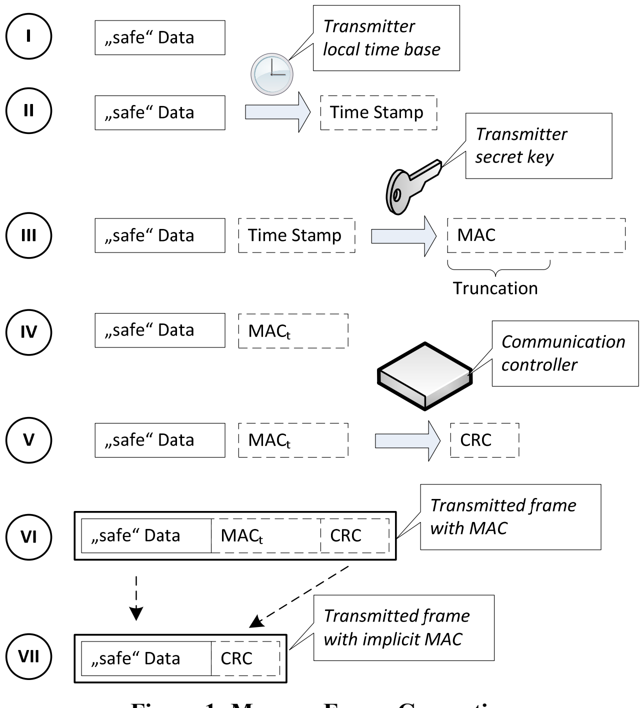

Figure 1: Message Frame Generation

When the data is available (I) a time stamp is generated (II) and a MAC is calculated based on data and/or time stamp (III). The time stamp will be masked to allow a time window complete from the encoding moment to the decoding moment (passing through the communication stack, network propagation, etc.).

After the time stamp is used in step (III) for generating the MAC it is no longer needed – the time information is embedded in the authentication code, protected with the secret key.

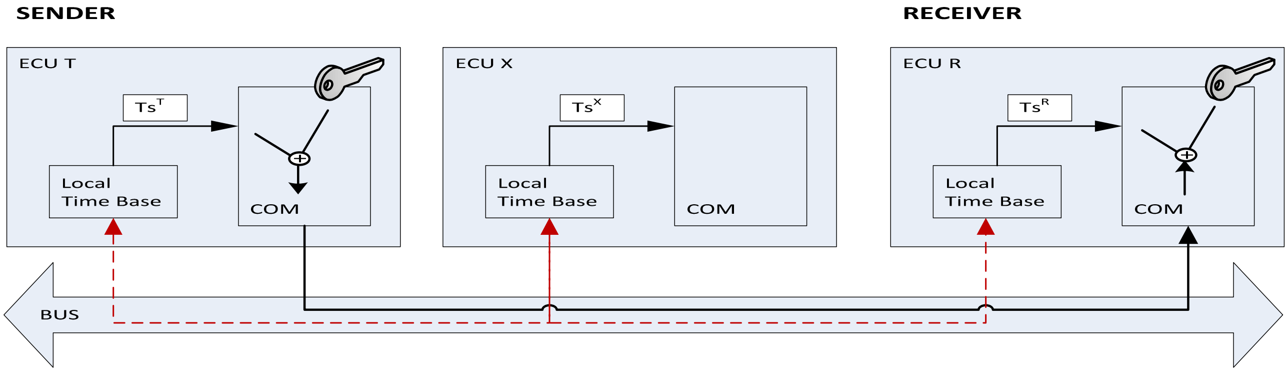

Figure 2: Network communication

In step (IV) the generated MAC (or truncated MACt) is added to the message payload and this will be propagated through the communication stack. The next stage (V) is the preparation for the physical layer transmission for which an extra redundancy is added (e.g. a CRC) in order to compensate for the physical transmission channel fault model (e.g. [7]).

Finally, the aggregated frame (VI) can now be transmitted on the network or a frame (VII) with omitted MACt for maximum data reduction. This data reduction is possible since the MACt is included in the CRC. The drawback of this approach is that the CRC check on receiver side can only be done after the MAC calculation. There are however compromises possible with a short CRCpl for the physical layer and a long CRCmac for the end to end protection.

To be noted that the inclusion of the integrity and freshness information in the CRC does not reduce its capabilities concerning the fault model of the physical connection. The reason is that the implicitly included MAC bits are not transmitted over the physical channel but instead calculated on the receiver side.

The receiver will follow the same reverse procedure for verifying the authenticity and integrity of the data. For this however, the receiver shall provide its own time stamp and add it to the received data in a step similar to step (III) in Figure 1.

This implementation assumes the existence of an infrastructure capable of distributing a secret key and a consistent time base to all the participants in the required communication. This is in general a normal assumption and standard SW infrastructure is in general available (e.g. [1]) for such distributed time bases. The precision / granularity of the time distribution mechanism will of course have to be in close correlation with the time window allowed by the coding mechanism for embedding the time stamps in the generated MAC (combined also with typical values for propagation across the SW stacks the communication controller, the bus, etc.).

One simplified view of the communication between ECUs on the network is described in Figure 2. The sender (ECU T), combines its local time stamp (T_s^T) with the secret key (pre-shared with the intended receiver, ECU R) into a MAC which will be part of the transmitted signal on the network (bus). The receiver, in this example ECU R, will combine into the received frame its own time stamp (T_s^R) and recalculate the MAC of the whole. If the received MAC and the locally calculated MAC match, there is a confirmation that the sender is authentic and the data received is integer and in the required expected “freshness”. In this way, the receiver will not only recognize an authentic sender but also the time interval in which the signal was transmitted. There is no way for another malicious ECU SW (e.g. ECU X in Figure 2) to successfully recreate a valid message at any other time without having access to the pre-shared key used by ECU T and ECU R.

A replay attack (for example an attack in which the ECU x is intercepting the message sent by ECU T and is trying to send the same message on the bus at a later time) will not succeed due to the wrong time stamp already embedded in the transmitted (original message) MAC; the receiver will fail to generate the same MAC with another (in principle a later one) time stamp and therefore will treat the message as non-authentic, non-integer or not fresh enough (each of these classes of errors being enough to reject the message).

A replay attack will work only in the allowed time window set up by the transmitter in the moment of the coding. There are in principle two options of dealing with this problem. If the replay attack is done in the allowed time window it usually will not do any harm except the overhead of the receiver being asked to process two times the same message. If this double message in the time window is not acceptable, a counter may be simply added in the protocol (one example is the end to end communication protection message counter [1]).

4.1 Use Case Example: Minimum Resources

The sender is for example a sensor which transmits periodically every ms a 32 bit value. For security a 16 bit truncated MACt and for safety a 12 bit CRC value are demanded. The time base range is 32 bit. So overall there are 60 bits of data. In the following the proposed solution is tailored for minimum computation resources and bus bandwidth.

The MAC calculation is purely based on the time value and can be done in advance having an execution time of up to 1ms. With these low demands a small (silicon area) and low power sequential crypto processing unit can be used. The bonding between sensor value and MAC is done with the CRC calculation. Two points need to be taken into account. Firstly the CRC will be extended to 16 bit for getting the protection of a 16 bit MACt, when the MACt itself is omitted (the 12 bit CRC was chosen to make this point). Secondly the truncated MACt, which is used for the CRC calculation has to be extended by the number of CRC bits since the CRC calculation is reversible. This means a 32 bit MACt is used for CRC calculation in this example.

The resulting message length is 48 bit with a 32 bit value and a 16 bit CRC which is significantly less than the original 60 bits. The 16 bits CRC carry all necessary information for authenticity, freshness and integrity checking from end to end.

4.2 Use Case Example: Legacy Compatibility

In a network of ECUs which partly do not have a crypto hardware but still should be able to read the content of messages, the integrity check and the transmission error CRC check need to be separated like for the frame structure in step (VI) in Figure 1. This is given per default, when the CRC is a mandatory part of the protocol specification on the physical layer like for CAN. But also in this case it is possible to reduce the computation requirements for the crypto unit very significantly, when the MAC(t) is purely (pre-) calculated based on the time value and a "CRCmac" is transmitted instead of the MACt.

5. CONCLUSIONS

The proposed solution addresses the needs for authentication, integrity and freshness in automotive network communications. Due to the increasing potential for malicious attacks using the multitude of attack surfaces in modern vehicles, such techniques may be used for protecting critical information propagated on the internal on-board network between the ECUs.

By applying the presented techniques, the authentication of the transmitter is also coupled with the time fingerprint of the message thus making reply attacks practically impossible.

Due to its significantly reduced performance requirements for the crypto hardware (MAC is pre-calculated with the time value) security capabilities can be introduced for components like low cost sensors, for which up to now this was too expensive. Or for a microcontroller more data channels can be protected with a crypto hardware of a specific performance class.

Another advantage is the reduction of bits, which are transmitted physically and through the different SW layers for an end to end protection. This allows introducing such a protection against malicious attacks also for bandwidth limited legacy bus systems.

REFERENCES

[1] AUTOSAR: Automotive open system architecture. http://www.autosar.org/

[2] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, Experimental Security Analysis of a Modern Automobile, 2010 IEEE Symposium on Security and Privacy

[3] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham,, Stefan Savage Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno, Comprehensive Experimental Analyses of Automotive Attack Surfaces, 2011

[4] IEC 61784-3 / 2009

[5] Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005, Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930NIST Special Publication 800-38B

[6] Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001, Federal Information Processing Standards Publication 197

[7] Controller area network (CAN); ISO 11898-1:2003, ISO 11898-2:2003, ISO 11898-3:2006