Signal-based Automotive Communication Security and Its Interplay with Safety Requirements

Benjamin Glas*, Jorge Guajardo†, Hamit Hacioglu*, Markus Ihle*, Karsten Wehefritz‡, Attila Yavuz†
*Robert Bosch GmbH, Stuttgart, Germany
†Robert Bosch LLC, Pittsburgh, USA
‡Bosch Engineering GmbH, Abstatt, Germany

***Abstract*—Recently, demonstrated attacks on automotive communication systems have made security a necessary requirement for future products. In this paper, we are concern with the problem of designing efficient integrity assuring mechanisms for highly constrained automotive (internal) network environments, such as for example the CAN bus standard. In particular, after briefly reviewing basic principles behind the design of challenge-response protocols, we discuss three possible protocols based on message authentication codes designed to guarantee data integrity in automotive internal networks. The three methodologies contrast different trade-offs which can be made during the design of such a system and that include: security, bandwidth overhead, and latency. We make particular emphasis in building our solutions on top of existing standards to allow for backward compatibility of the presented solutions. Thus, our proposed solutions are practical and readily deployable.**

I. Introduction

For decades safety has been one of the key properties of automotive subsystems, especially for electronic systems, which have steadily gained importance in the industry. Recently rising awareness and demonstrated attacks [1], [2] on automotive communication systems have made security a necessary requirement for future products. These recent works show that direct access to vehicle internal bus systems is possible in many different ways with reasonable effort. In addition to directly tapping physical bus lines, the On-Board-Diagnosis (OBD) interfaces can be used to eavesdrop on the internal communications of a modern automobile. Wireless interfaces, such as Bluetooth and cell phone connections offer possibilities for remote attacks. Also, software in some bus-connected ECUs is vulnerable to manipulation and compromise, so that these bus participants can be hijacked and leveraged to mount attacks on the automobile internal communication network.

In practice, attackers might be motivated to manipulate safety critical systems directly to deliberately cause harm or for economical gain. For example, manipulation of sensor or control data can be used to enhance performance (e.g. tuning of engine) or to enable functions outside their safety perimeters (e.g. operate convertible top at higher speeds). Since this data is often used to feed complex control circuits or for several, sometimes safety-critical functions, these manipulations might have additional impact unintended by the attacker. Therefore authentication and integrity protection of sensible data is necessary to protect correct and safe functionality of the vehicle systems.

In classical automotive communication buses such as CAN, communication is widely signal-based with short data snippets broadcast to a set of receivers. In these systems, communication is highly constrained by real-time and latency requirements, already high bus loads and strict cost limits. This distinguishes embedded automotive systems from traditional broadcast networks (e.g., the Internet) with similar security problems and goals (authenticity, integrity, confidentiality, etc) but with very different resource availability and complexity. Thus, security mechanisms for automotive systems have to be tailored to be resource efficient with respect to processing and communication overhead while, at the same time, protecting safety critical communications.

This contribution is concerned with automotive in-vehicle communication and, in particular, with the CAN protocol and its (in)security. We propose resource-efficient protection mechanisms for critical signals that can be applied to current communication systems and architectures with minimal impact to allow protection even of legacy systems. We see application-specific solutions for safety-critical sub-systems as first immediate steps to a more holistic future secure communication, e.g. based on IP-based communications (see e.g. SEIS project [3]). As an example we look at Adaptive Cruise Control (ACC), communicating brake requests to the Electronic Stability Program (ESP) and torque requests to the engine management over CAN buses.

The remainder of this contribution is organized as follows. Section II begins by presenting our assumptions and the constraints under which propose solutions. We also briefly recall security primitives and design approaches, which will be used in later sections. In Section III, we present three possible approaches to solve the problem of guaranteeing integrity in automobile communication internal networks. We consider the interdependencies between safety and security. We then analyze the implications of such interdependencies on mechanisms aimed at providing reliability, efficiency, and protection against malicious attacks and manipulation in an automotive platform. We highlight advantages and disadvantages of each approach. At the end, we present a method to prevent attackers from easily mounting denial of service attacks in the internal networks of the car.

II. Preliminaries

A. Model and Assumptions

We assume the internal bus systems of a modern automobile to be insecure in the sense that a potential attacker has full access to the communication network in a classical Dolev-Yao attacker model [4]. In particular, this means that the attacker can freely insert, delete, manipulate, or delay any messages on the bus. In practice existing error correction and monitoring mechanisms of the field buses reduce the impact of some attacks since simple manipulations are detected as communication errors, typical of the stressful environment of a car. We assume that the attacker has no access to protected key material and that cryptographic operations are computed correctly (i.e. secure processing environment). This can be guaranteed via a Hardware Security Module (HSM), for example. Furthermore, we assume standard security practice of using different keys for different cryptographic operations (e.g., encryption and authentication) is followed. In addition, we assume that cryptographic mechanisms used (e.g. AES, SHA-2, etc.) are secure. As it is traditional in the cryptography literature, we will assume two honest communicating parties, often referred to as the prover (also Alice) or sender and the verifier or a receiver (also Bob) and an attacker (also called Oscar or Eve).

We assume a typical driving cycle in the automotive area, consisting of a start-up or boot-up phase that has to be as short as possible and an usually longer driving phase with normal (continuous) operation of the system. In the start-up phase the sender has to authenticate itself to the receiver. This is done as the first step of the security protocol, which can be repeated during operation as needed (e.g. on a timer event or a counter overflow). During the driving phase, data integrity has to be guaranteed, which is done by providing authentication data to the receiver in addition to the transmitted (data) payload. This is the second step of the protocol which runs continuously during operation.

Finally, although security during production, logistics, and maintenance all impact directly the security of the automobile internal network and it is challenging to guarantee, in the current work, we do not concern ourselves with the processes or technology required to guarantee security in these environments. In particular, all such processes are out of the scope of the current paper.

B. Security Primitives and Building Blocks

The main security property that has to be guaranteed in automotive internal networks is (unilateral) authenticity of the sender entity and the data sent. Notice that data authenticity implies data integrity, since data that has been manipulated is not authentic any more. For the sake of clarity we will refer to authentication when talking about entity authentication and about integrity, when dealing with data integrity, always implicitly meaning data authentication. In what follows, we provide an overview of cryptographic primitives that can be used to guarantee entity authentication and data integrity and compare their advantages and disadvantages.

Observe that a basic requirement to be fulfilled to provide both entity and data authentication it has to be proven that the valid sender of the data is alive and present, genuine, and actually originator of the data received. The next two sections describe known mechanisms that can be used to achieve entity authentication and data integrity.

1) Entity Authentication and Freshness: For authentication the verifier wants to be assured that he is communicating with the intended prover or sender. It is well-known [5] that entity authentication can be achieved via possession of a physical token, knowledge of a secret (password or a key), ability (e.g. to do a transformation), or an intrinsic property of the prover (e.g. biometrics or a physically uncloneable function) always under the assumption that only the valid prover has access to that credential or property. Additionally the protocol has to assure freshness, meaning that the prove cannot be computed in advance and recorded and replayed by an attacker. Freshness can be achieved via challenge-response protocols. In particular, in a challenge-response protocol, the verifier sends a challenge (usually a random number, that is used only once, also called a nonce) to the prover, which in turn computes a response based on the challenge and its secret credential (key) which can then be verified by the verifier. Because the challenge is chosen by the verifier, the response cannot be precomputed or replayed by the prover.

The approach has two major drawbacks for resource-constrained embedded systems: It needs bidirectional communication and at least two messages for challenge and response, creating overhead and latency. Possible alternative approaches include timestamps and strictly monotonic secure counters. Unfortunately, both of these solutions are very hard (costly) to realize on small and simple embedded devices such as sensors in the automotive domain. Since we look at use cases on the (bidirectional) CAN bus, the approach described in this paper is based on a challenge-response protocol to achieve entity authentication.

2) Symmetric versus Asymmetric Approaches: Authentication and integrity are security properties that have some asymmetry in nature: The sender only has to prove, the receiver only has to verify. Therefore, asymmetric cryptography approaches such as digital signatures should be evaluated as a potential avenue for solutions. Asymmetric cryptography approaches have several advantages, which include: (i) key management is easier, public keys do not have to be kept confidential (only integrity-protected) or alternatively, a public-key infrastructure must be present to allow certificate validation, (ii) the risk of key compromise is minimize as private keys do not have to be shared between multiple parties and secure communications can be easily defined bilaterally, and (iii) scalability is optimal in asymmetric cryptography systems, since users only need one public-key (thus, the complexity of the system grows linearly in the number of participants). Unfortunately, in the context of embedded systems (as those found in a modern automobile), asymmetric solutions have three major drawbacks:

- Computational complexity is much higher for asymmetric

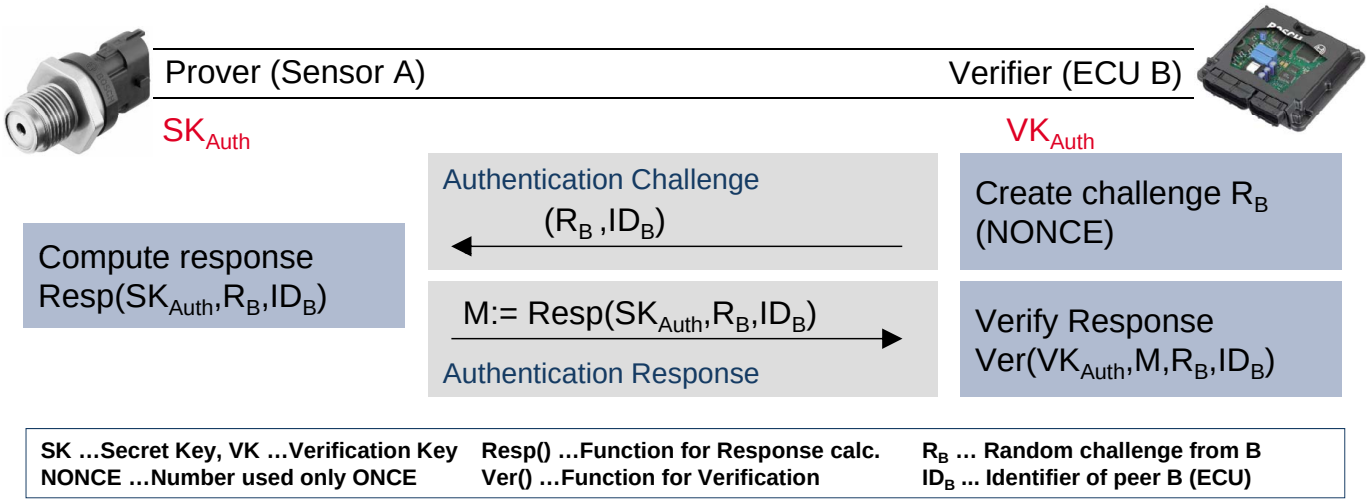

Figure 1: Unilateral authentication via challenge-response protocol

systems, resulting in long processing times on the nodes and/or high implementation effort and code size.

- Key (both private and public) are longer in general in comparison with symmetric systems, resulting in costly storage requirements and longer blocks for processing and transmission.
- Finally asymmetric mechanisms such as signatures cannot be truncated for transmission, since the verifier needs to know the complete signature for verification, thus, resulting in higher communication overheads than comparable symmetric cryptosystem-based solutions.

The previous disadvantages make symmetric-key based approaches much better suited to the constrained environment of the internal network of the modern automobile. Thus, in the remainder of this paper, we will be concerned with protocols based on symmetric-key primitives and in particular, message authentication codes or MACs. Message authentication codes have many advantages, which include: high security level depending on the key size and MAC size used, they can be implemented compactly and efficiently in software and in hardware, and they can be truncated for transmission. The last property is very important in our setting as it implies that the MAC length can be adjusted (truncated) and traded off as a parameter against communication overhead (at the cost of an increase probability of forgery via collisions).

III. Achieving Authenticity in Automotive Intra-Networks

In order to achieve authenticity and freshness we propose using a simple challenge-response protocol based on a pre-shared symmetric key and message authentication codes as a basis for entity authentication. In order to achieve integrity of the data transmitted during the driving phase, we propose flexible usage of MACs based on symmetric block ciphers (e.g. CMAC [6] based on AES [7]) and inclusion of counters to assure freshness. Main problem on the communication side is the overhead caused by the additional data in combination with possible additional latency. Both are especially challenging when dealing with short signals requiring real time operation and low latencies. Depending on the application needs two general approaches seem possible.

A. Message-centric Approach: One MAC per Message

Our first alternative is the classical approach to add a MAC to every single message. Since the maximum payload of a CAN message is only 8 bytes which has to comprise both payload data and MAC, both contents are very limited in length which puts very harsh limits to the security of the single MAC even with only some few bits of data. To mitigate these limitations, two approaches are looked at: Action trigger groups and combination of safety and security measures.

The first leverages the fact, that in most applications the triggering event for a reaction of the system does not consist of a single message but of a group of messages that have to be received correctly. This can be a handshake between ECUs, some sensor values that have to be received in a row that are over a certain threshold or even more complex sequences. This series of messages and events we call an "action trigger group". If implemented properly, a possible attacker has to fake the complete group of messages to achieve the intended system reaction. Therefore, the security evaluation can look not only at the security (or insecurity) of a single message, but at the complete group. The probability of an attacker faking the complete group of messages in a row can be lowered below a desired threshold even with single message MAC lengths that would alone not be considered sufficiently secure.

The second mitigation approach is using synergies between safety and security measures. When looking at integrity of messages, both safety and security aim at the same goal - providing guarantees that the receiver gets the very same message that the sender intended to send. Only the adversaries are different: natural and usually random errors on the safety side versus a deliberately acting, malicious attacker on the security side. We observe that security primitives such as

cryptographic MACs have properties that can also be used by safety applications, since a transmission error of a single or multiple bits is detected by a MAC with a very high probability (depending on the MAC length). Assuming a secure MAC of length L bits, the probability that an error (independent of the number of erroneous bits!) is not detected is 2^{-L}. The main observation is that the MAC tag can be used for error detection *and* for integrity protection. Therefore, the communication overhead imposed by adding a cryptographic MAC can be reduced by the size of existing error detecting codes (e.g., CRCs) already included at the application level.

B. Data-centric Approach: Decoupling MACs and Messages

A second alternative is the decoupling of messages and MACs with possibly independent transmission. This means that for a certain set of data (such as sensor values) a MAC is computed which is transmitted in parallel with or after transmission of the data. Since all data of the set is needed to compute the MAC, depending on the data set size, some delay occurs between reception of the first piece of data and final verification of the MAC (including the very last bit of data of that set). If the system can cope with some uncertainty regarding the authenticity of the data, all data can be sent and used immediately on the receiver side, but validation of authenticity can only be done when the complete set of data and the MAC have been received. Figure 2 shows a schematic overview.

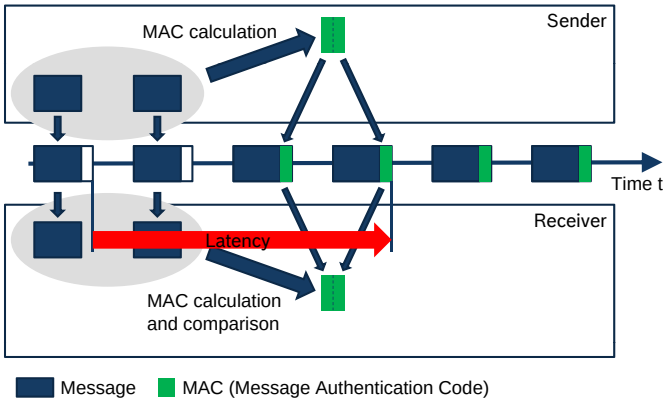

Figure 2: Schematic view of MAC slicing and lagged transmission

It's apparent from Figure 2 that the MAC is calculated from a set of data, eventually distributed over several messages. After calculation this MAC can be sliced into an arbitrary number of parts and transmitted with the different blocks of data, e.g. attached to messages carrying the next set of data. Since the validation is done only after receiving both the data and all MAC parts, there is some latency before verification, which can naturally fail. Therefore the system is at risk of using non-authentic, manipulated data and only detecting it afterward. On the other hand, this approach enables distributing the overhead over a larger set of data and therefore using larger MACs even if only very little bandwidth is available. A use case for this approach would be e.g. tuning detection, when tuning shall not be prevented but detection is sufficient. Here some latency (e.g. some milliseconds) can be accepted since the system reaction (logging, signaling, safe mode) does not have to be immediate.

When decoupling MACs from messages and grouping data from different messages together, synchronization of prover and verifier is an issue. If single messages are lost or doubled on the channel or during routing, the MAC verification fails. This has to be considered in the protocol design. For some safety critical systems, alive counters are integrated on application level which could be reused for detecting message losses or doublings. Alternatively the transmission of MACs could be embedded in a additional protocol to detect errors. Unfortunately, this introduces additional overhead reducing the possible security level assuming constant bandwidth availability.

C. Signal-based Approach: Individual MACs for Single Signals

The data-centric approach can also be used to attach MACs to single signals instead of complete messages. In CAN communication often several signals (of only a few bits each) are put together in one CAN frame to save bandwidth. But since these signals may be interesting for different receivers, the frames are not necessarily forwarded as a whole when passing through a gateway, but individual signals may be regrouped or routed independently. Therefore a MAC for the whole frame is no longer verifiable at the final receiver. Here the data-centric approach with sliced MACs would enable including MACs for single signals. So to each signal value a MAC slice could be attached that is routed as part of that signal. At the final receiver the MAC slices are put together again to verify the integrity of this signal values.

IV. System State Hiding Via Encryption

Finally we look at an additional attack method for safety-critical systems: selective Denial-of-Service (sDoS). On a wired bus-like communication system, an attacker with physical access can usually perform a denial-of-service (DoS) attack by simply spamming the channel, in the CAN case e.g. by applying the dominant voltage value permanently. In practice, this is detected by the safety system as a communication error and the system is put in fail-safe mode. The fail-safe mode, however, often leads to the de-activation of the respective system. If the attacker was able to precisely detect when the system is in a critical situation (requiring a real-time and immediate response), the attacker could mount a selective DoS attack at exactly this point in time and lead to dangerous and potentially unsafe situations (since the driver would get critical safety information too late to take effective action). Therefore, it would be beneficial to hide the system state from an attacker to prevent him from determining the right point in time to perform a sDoS attack. One simple approach to prevent the attacker from seeing control information is to encrypt the control and sensor data. Since constraints of the systems remain unchanged — short messages, hard real

time requirements, little available bandwidth — a standard approach with block ciphers of reasonable length (e.g. 128-bit AES in CBC mode) would result in an unacceptable overhead, if we chose to encrypt each data block sent as a 128-bit encrypted block. Therefore, we recommend using a suitable stream cipher, e.g. a block cipher in counter mode. Thus, by combining encryption in CTR-mode with one of the integrity protection schemes[1] suggested in the previous section, one could achieve confidentiality and authentication protection with no significant additional transmission overhead as long as the content of the message cannot be deduced from the length of the message.

V. Conclusion

In this contribution we analyze requirements and security goals and propose concrete solutions to the problem of authentication and integrity protection in automotive internal networks. We describe different approaches and compared them according to their costs (performance, bandwidth, latency) and security properties. Depending on the concrete use case and application, a particular solution can provide a tailored level of security while keeping system impact and overhead minimal.

References

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," *Security and Privacy, IEEE Symposium on*, vol. 0, pp. 447–462, 2010.

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2028067.2028073

[3] Sicherheit in Eingebetteten IP-basierten Systemen (SEIS) project, "Homepage of the seis project," http://www.strategiekreis-elektromobilitaet.de/public/projekte/seis, 2012.

[4] D. Dolev and A. C. Yao, "On the security of public key protocols," in *Foundations of Computer Science, 1981. SFCS '81. 22nd Annual Symposium on*, oct. 1981, pp. 350 –357.

[5] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001. [Online]. Available: http://www.cacr.math.uwaterloo.ca/hac/

[6] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication," U.S. Department of Commerce, Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, NIST Special Publication 800-38B, 2005.

[7] FIPS, "Pub 197: Advanced Encryption Standard (AES)," U.S. Department of Commerce, Information Technology Laboratory (ITL), National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Federal Information Processing Standards Publication, 2001, electronically available at http://www.itl.nist.gov/fipspubs/.

[8] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Advances in Cryptology - ASIACRYPT 2000*, ser. LNCS, T. Okamoto, Ed., vol. 1976. Springer, December 3-7, 2000, pp. 531–545.

[9] H. Krawczyk, "The order of encryption and authentication for protecting communications (or: How secure is ssl?)," in *Advances in Cryptology - CRYPTO 2001*, ser. LNCS, J. Kilian, Ed., vol. 2139. Springer, August 19-23, 2001, pp. 310–331.

[1]The subject of combining encryption and authentication schemes has been studied previously and it is well-understood. See e.g, [8], [9].