SECURITY SOLUTIONS FOR THE CONTROLLER AREA NETWORK

Bringing Authentication to In-Vehicle Networks

IMAGE LICENSED BY INGRAM PUBLISHING

Bogdan Groza and Pal-Stefan Murvay

Vehicles cannot be secured while the in-vehicle network remains insecure. The growing number of attacks reported each year shows that in-vehicle buses are not isolated from the outside world. By exploiting their lack of security, adversaries can gain control over virtually any functionality inside the car. We discuss the most promising approaches for assuring security on the controller area network (CAN) bus after a decade of attacks and security proposals. Most of the proposals are based on cryptographic mechanisms, but some exploit the physical layer or even the physical characteristics of the controllers. The surveyed solutions prove a significant degree of maturity and sophistication, which suggests that the moment for adoption and standardization by the industry has come.

Digital Object Identifier 10.1109/MVT.2017.2736344
Date of publication: 25 January 2018

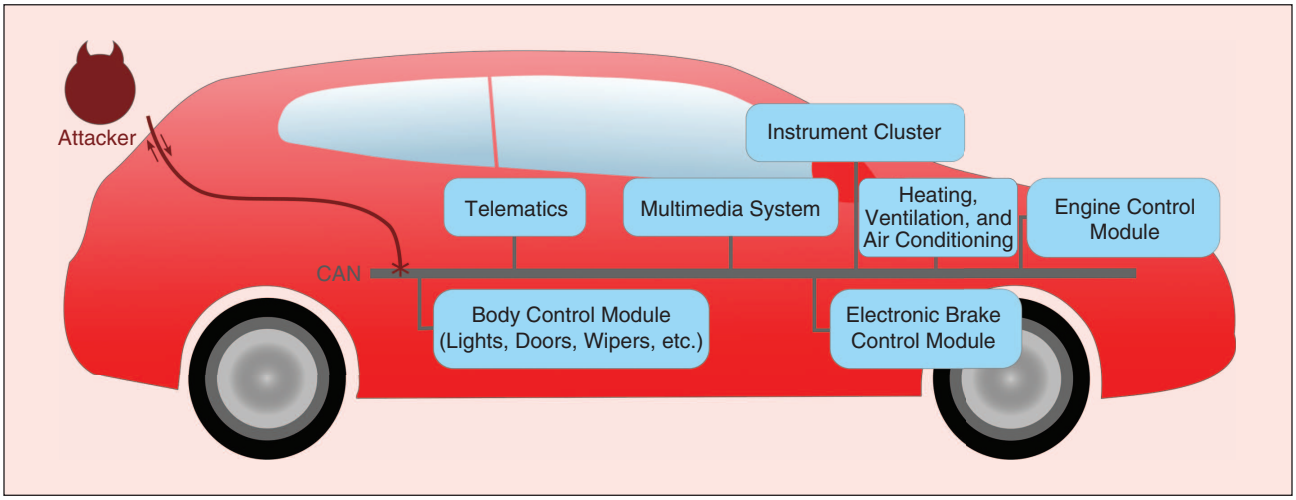

FIGURE 1 The automotive modules attacked via the CAN bus.

The CAN Bus as an Attack Surface

Security through isolation has always been an illusion. In the era of cyberattacks, complex incidents such as the Stuxnet worm proved that even isolated facilities cannot remain secure from attacks orchestrated by strong-willed outsiders. Cars are no exception.

The first attack on in-vehicle networks that we could trace involved the completion of a rather mundane task: playing with the electric window lift [5]. Just a few years later, the first comprehensive analysis of in-vehicle security [7] demonstrated the corruption of various modules of a real-world car, including safety-critical components such as the engine control module, the brake control module, and the body control module. Currently, dozens of attacks are disclosed each year in research publications or through the news. A practical survey of in-vehicle vulnerabilities can be found in [10].

The vast majority of the attacks were launched through the in-vehicle network, e.g., the CAN bus, which mediates access to all of the existing system components and functionalities. Access to the in-vehicle network can be achieved through the diagnosis connection or by directly tapping the bus wires; in the case of remote connectivity, this can even be done from the outside. Once access to the bus is established, virtually all modules employing CAN-based communication are prone to attacks; the adversary can lock the brakes, steer the car, kill the engine, and virtually control the car at will. This view is suggested in Figure 1.

The CAN bus is the workhorse behind most of the existing in-vehicle networks. It was designed by Bosch in 1983 and proved to be so successful that, today, it is present in every car on the market. Newer alternatives such as FlexRay or BroadR-Reach (an Ethernet-based technology) bring more bandwidth but at a higher cost without improving security. Thus, most attacks reported on the CAN bus are valid in all other existing in-vehicle network embodiments and, more importantly, the countermeasures proposed for CAN are largely extensible to all other in-vehicle buses.

The CAN bus is a two-wire broadcast bus, as suggested in Figure 2. At most, 64 b of data can be carried by one frame. Recently, CAN with a flexible data rate (CAN-FD) was introduced as an alternative to CAN. It allows a higher data rate during the transmission of the data field, which can be extended to 512 b (64 B). This makes CAN-FD more suitable for security at the application layer, which incurs higher payloads for the messages. The CAN bus was designed with reliability in mind; a 15-b cyclic redundancy check that accounts for transmission errors is simple and effective, but it has no intrinsic security mechanism.

A Historical Perspective

The increased connectivity is what opens doors to outsiders. In Figure 3, we depict several steps in the evolution of some security threats and defense mechanisms from the computer industry. From the development of the Advanced Research Projects Agency Network (ARPANET), a foundational brick of today's Internet, it took a bit more than a decade for the first computer viruses to

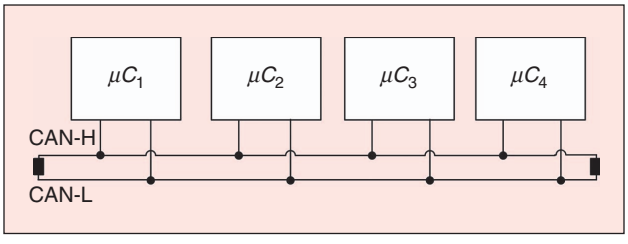

FIGURE 2 The topology of the CAN bus. CAN-H: CAN-High; CAN-L: CAN-Low; μC: microcontroller.

appear and the first defense mechanisms, firewalls, and antiviruses to be put into place. In the 1990s, the need for cryptographic security became obvious, and the development of the first cipher suites started; e.g., Internet Protocol security and secure sockets layer/transport layer security. Today, we cannot imagine the Internet without these security mechanisms, and security through isolation is not an option in the computer industry.

Cars will likely follow the same path, especially as remote functionalities for diagnosis, software updates, vehicle access, and similar functions become a must. If in-vehicle networks follow computer networks a decade or two later, then we can expect the first in-vehicle malware to arrive in the next few years. This clearly leads to the need for devising security for in-vehicle buses, in particular, for the CAN bus. Some of the academic research efforts related to creating security mechanisms for the CAN bus are summarized in Figure 4 in a potential chronological order.

Adoption by the industry requires standardization. The automotive open system architecture (AUTOSAR) standard has included specifications for cryptographic support since version 4.2.2 in 2015. Still, for the moment, there is no standardized cryptographic protocol for in-vehicle buses.

In-Vehicle Electronic Control Units are Ready for Cryptography

When it comes to performance, in-vehicle electronic control units (ECUs) have kept up with the increasingly demanding industry requirements. Coping with the demands of cryptographic algorithms comes within reach for more devices as operating frequencies and memory sizes continuously grow.

To illustrate the capabilities of automotive microcontrollers in handling cryptography, Table 1 presents the execution speed for several cryptographic functions on four automotive-grade devices using an 8-B input (i.e., the maximum payload of a standard CAN frame). Four representative cores are taken into account, coming from both low- and high-end platforms: 1) the Freescale/NXP S12XD, a member of the Freescale S12 family used in various powertrain, chassis, and safety applications; 2) the Freescale/NXP MPC5606B, a general-purpose automotive microcontroller; 3) the Infineon TC1797, a powertrain, chassis, and safety applications controller; and 4) the Texas Instruments TMS570LC457, a microcontroller designed for automotive safety-critical applications featuring a Cortex-R5 ARM core.

The results show that encryption algorithms such as the advanced encryption standard require a processing

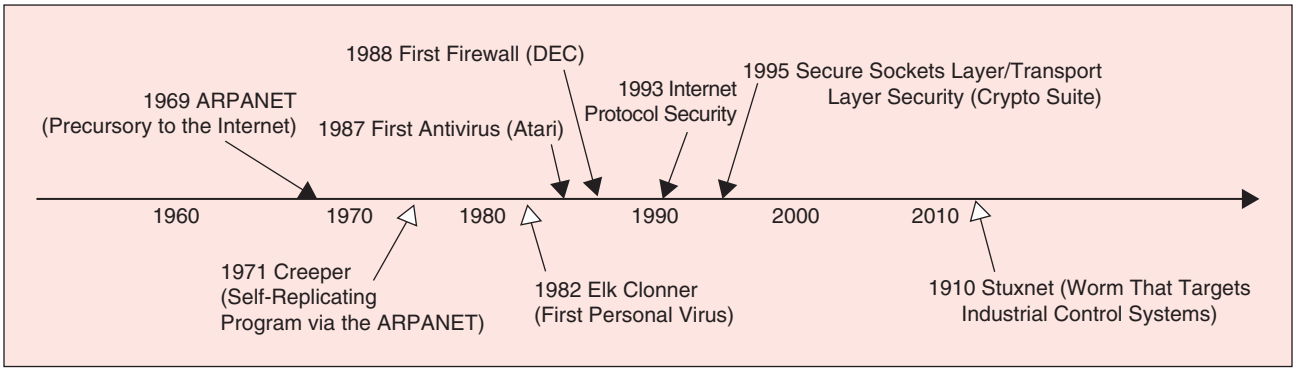

FIGURE 3 The evolution of some security threats and defense mechanisms in the computer industry. DEC: Digital Equipment Corporation.

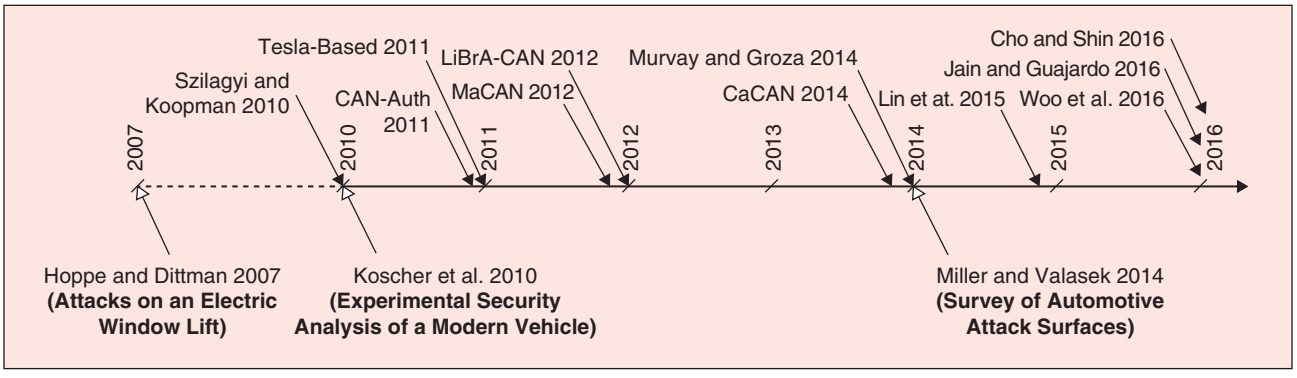

FIGURE 4 The protocol proposals for assuring CAN bus security in potential chronological order by publication year, following the first reported attacks on in-vehicle networks. LiBrA-CAN: lightweight broadcast authentication protocol for CANs; MaCAN: message-authenticated CAN; CaCAN: centralized authenticated CAN.

time on the order of dozens of microseconds on high-end platforms and hundreds of microseconds for the low-end ones. Similar results are achievable for hash functions such as Secure Hash Algorithm 1 (SHA1) and SHA256 or message authentication codes (MACs), which proves that they are suitable for practical applications. The newer SHA3 standard has a somewhat poorer performance but is still within reach. We exclude SHA3 from the graphical depictions in Figure 5 to allow a clearer comparison between the remaining candidates. Additionally, in Figure 5, we add the computational time on 64-B inputs, i.e., the size of a CAN-FD frame.

Cryptographic Security at the Application Layer

Using cryptography at the application layer is the most natural choice and is in line with what is already done in computer networks. Due to limited bandwidth and existing computational constraints on vehicular ECUs, standard MACs are the cryptographic function of choice for assuring message authenticity.

MACs require a secretly shared key between the participants. The MAC function is applied over the message and the secret key to generate an authentication tag that is verified by recomputing it based on the received message and the known secret key. The hash-based message authentication code (HMAC) is the most popular choice for a MAC. It requires applying a cryptographic hash function twice along with a secret key k and two constant padding values, *ipad* and *opad,* on the message:

$$HMAC(k, message) = Hash(k \oplus ipad \parallel Hash(k \oplus opad \parallel message)).$$

While most of the protocol descriptions that follow employ regular MACs, the difference between them is in the way the secret keys are shared and/or used for computing the MACs.

TABLE 1 *Computational performance on automotive-grade controllers on various cryptographic primitives with 64-b inputs.*

Platform	SHA1	SHA256	SHA3-256	HMAC-SHA1	HMAC-SHA256	HMAC-SHA3	AES
S12XD	856 μs	1.754 ms	56 ms	3.815 ms	6.950 ms	113.4 ms	663 μs
MPC5606B	958 μs	604 μs	29.7 ms	3.935 ms	2.350 ms	63.1 ms	934 μs
TMS570LC	76 μs	83.8 μs	2.21 ms	376 μs	347 μs	4.815 ms	142 μs
TC1797	48.6 μs	57.7 μs	5.23 ms	213 μs	223 μs	10.62 ms	55.7 μs

HMAC: Hash-based message authentication code.

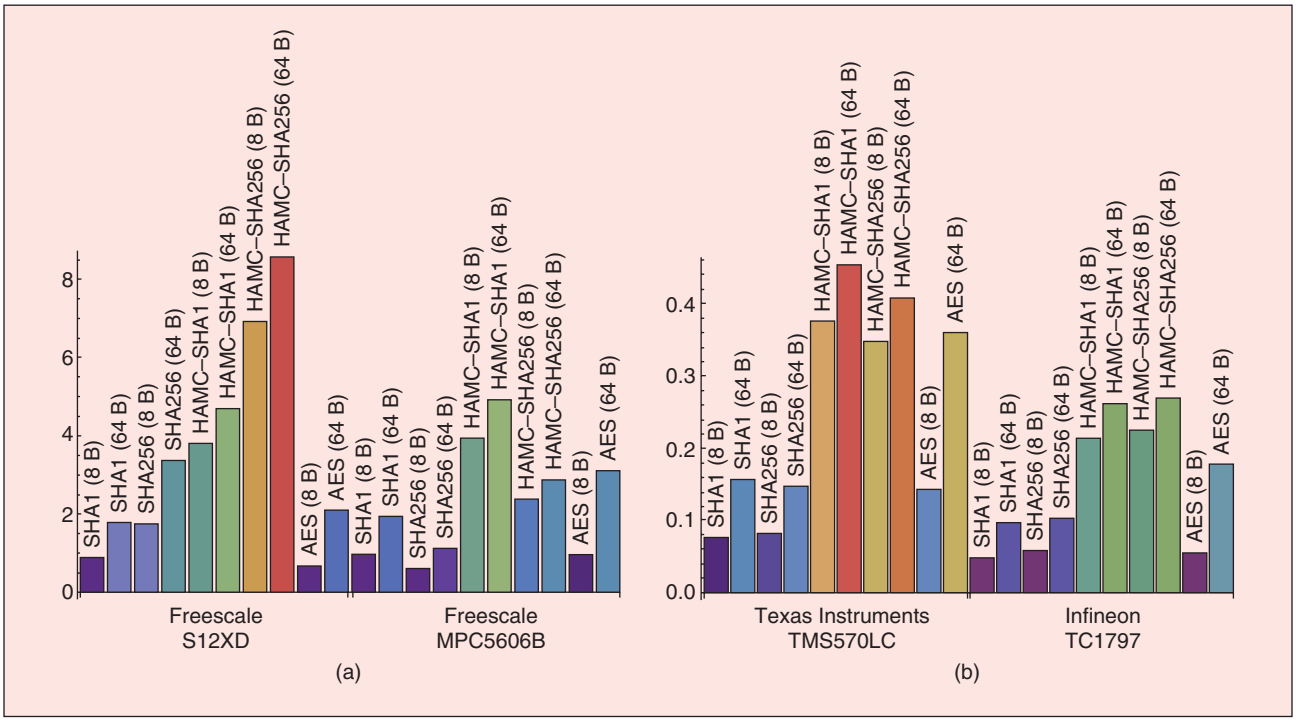

FIGURE 5 The computational time for (a) low-end and (b) high-end cores.

Regular Key Sharing and MACs

A unique secret key for the entire CAN network is not a good alternative because, if a single node is corrupted, then security is lost for the entire bus. The most obvious procedure is sharing secret keys in a pairwise fashion between nodes. This mechanism is employed by the following schemes and suggested in Figure 6(a), where each ECU holds one secret key shared with each of the other ECUs.

Voting schemes for time-triggered communication were introduced by Szilagyi and Koopman in [13]. The scheme is intended for the generic time-triggered communication such as that present in TT-CAN and FlexRay. Due to the limited space, the tags are truncated; three MACs, each 8 b, are fitted at the end of a single frame in a case of three receivers. Since each frame carries only a small amount of authentication information, a message needs to accumulate a sufficient number of votes (authentication tags) to be deemed authentic. The idea of voting does not really seem suited for the nature of the CAN, as the real-time nature of communication doesn't allow enough time for nodes to cast votes. Voting for past received messages is also suggested [13], but nodes may not share the same receive history due to identifier (ID) filtering at the hardware level or because nodes may go into a bus-off state. This proposal may be limited in application to the CAN bus, but it is the first research effort.

Message-authenticated CAN (MaCAN) is proposed in [4]. The protocol employs shared keys between nodes and MACs derived from block ciphers, i.e., the cipher-based message authentication code construction. This saves some of the computational time, as block ciphers such as the AES can be used. To cope with the limited size of the data field, MACs are truncated to 4 B. The authors of MaCAN [4] also suggest that nodes can be grouped under the same key if they share the same trust level, but no practical insights are given on how to decide the trust level. MaCAN uses the straightforward way of sharing keys between nodes and fitting a truncated MAC in half of the data field.

A more recent MAC-based scheme accompanied by experimental results for the newer CAN-FD can be found in [15].

Group Key Sharing

The main limitation of pairwise key sharing is that the limited space of a single tag is equally split between the receivers. For example, if the MAC space is limited to 24 b and there are three tags/receivers, each will get only 8 b of authentication. This is clearly too low, but more can be done than the simple pairwise sharing of keys between nodes, and more efficiency can be gained from a single authentication tag. The simple contrast between pairwise key sharing and group key sharing in Figure 6 shows why this is the case. In Figure 6(b), each ECU gathers the other ECUs into groups of three. Key *k11* is shared by ECU1 with ECU2 and ECU3, key *k12* with ECU2 and ECU4, and so on. When sending a MAC with the three keys, i.e., *k11*, *k12*, and *k13*, each of the three other ECUs will be in possession of two keys out of the three, thus gaining 16 b of the tag rather than 8 b. Therefore, the security level is doubled. In case of a single corrupted node, the security drops to 8 b, and if two corrupted nodes exist, the security drops to zero. Since in-vehicle networks are built by reputable manufacturers, corrupted nodes must be the minority. Group key sharing has considerable security advantages in cases when adversarial nodes are in the minority. This is exactly the principle behind the next proposal.

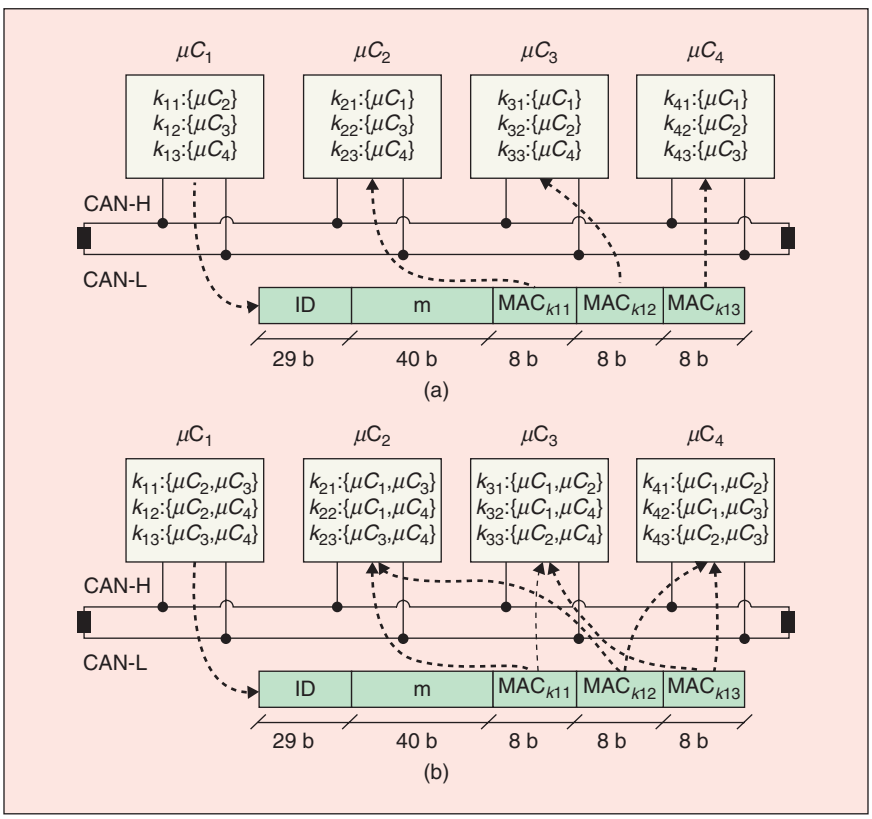

FIGURE 6 Pairwise key sharing (a) between nodes versus (b) group key sharing, as proposed in LiBrA-CAN. m: message.

The LiBrA-CAN [2] is the first to propose a group key allocation

procedure, which mixes keys between groups of nodes rather than sharing them in a pairwise fashion. Besides mixing the keys between groups of nodes, LiBrA-CAN also makes use of a more advanced MAC construction, which mixes the authentication tags, allowing forgeries to be detected even if they are done for another key of the same mixed MAC. LiBrA-CAN is a more demanding security protocol for the CAN bus, less straightforward to implement, but with certain security advantages in front of the other solutions.

Time Synchronization

The Timed Efficient Stream Loss-Tolerant Authentication (TESLA) protocol [12] was a breakthrough in sensor networks. This protocol allows symmetric keys to be used for broadcast authentication by releasing them in a time-dependent fashion. It enables simple, cost-efficient symmetric cryptographic functions to be used without secret keys for assuring broadcast authentication to multiple receivers. Bringing this protocol to the CAN bus was considered in [3], where several tradeoffs were studied. One drawback in adopting this protocol for in-vehicle networks is that it achieves authentication with a small delay as keys are released at fixed time intervals. This delay would usually be in the order of 1–10 ms, which may be small enough to preserve the real-time nature of CAN. The results in [3] prove that such protocols are within reach for in-vehicle networks.

Efficient Signal Allocation

Clearly, security mechanisms add overheads that can impede system performance. Lin et al. [9] discuss how to deal with both security and safety constraints. The authors consider tasks on each ECU as a source or destination for signals and a path as being an interleaving sequence of tasks and signals. Path-based security constraints are then formulated and a heuristic algorithm is proposed to efficiently find a solution to this problem. While this line of work does not come with a new security protocol (it relies on regular MACs and shared keys), the constraint-based signal allocation can be applied to any of the previous mechanisms for increasing their efficiency.

Security at the Physical Layer

Sharing the Keys

How to share keys between ECUs is a relevant issue. Standard cryptographic mechanisms, e.g., public-key mechanisms, can be used for this but come with high overheads. A novel mechanism is explored in [6], with keys exchanged in a secure manner by exploiting the physical properties of the signal on the CAN bus. Briefly, since dominant bits overwrite recessive bits, two nodes that send a message at the same time can still ascertain part of what the other has sent. This is further explored by [6], with a tree-based key-sharing mechanism, where

LiBrA-CAN is a more demanding security protocol for the CAN bus, less straightforward to implement, but with certain security advantages in front of the other solutions.

the leaf nodes of the tree are the physical nodes, while all the other virtual nodes in the tree correspond to logical entities that can be emulated by any physical node connected to it. The results presented in [6] evaluate the cost for M nodes to share an n bit key, showing certain advantages for the tree-based scheme.

Obstructing Forged Frames in the Physical Layer

Centralized authenticated CAN (CaCAN) [8] introduces a centralized view over the authentication process. In this protocol, a central node verifies the authentication tags of each frame, and if authentication fails, the frames are discarded with error flags. This procedure has the merit of requiring a single monitor node with higher computational power for this purpose. However, an adversary that removes this node from the bus can take full control of the network since there is no way for the other nodes to decide whether a frame is authentic or not. This may be a serious limitation for practical deployments and, likely, the only one that may stop CaCAN from becoming an industry-standard solution.

Hiding Authentication Bits with CAN+: CAN-AUTH

The Backward Compatible Broadcast Authentication Protocol for CAN (CANAuth) proposes the use of an ID-oriented key allocation procedure [14] and a nonstandard CAN derivative called CAN+. From a cryptographic perspective, CAN-Auth relies on simple MACs and shared keys. We mention it here because, while the cryptographic constructions behind it bring nothing new, the idea of using a new physical layer makes it of significant interest. CAN+ allows additional bits to be inserted during the transmission of a single bit. Concretely, as the sample point for the CAN bus is at around 75% of the bit-time, the CAN+ transmission window is placed at between 15–55% of the nominal bit time. This allows for authentication bits to be sent in a stealthy manner, making the solution backward compatible. Convincing experiments are provided on a field-programmable gate array implementation in [14]. The main limitation of this solution is the uncertainty regarding the practical adoption of CAN+. With CAN-FD already released, it seems unlikely for CAN+ to become a successor of CAN.

Security by Physical Characteristics

Physical characteristics of electronic devices open a new vista for security applications. In recent years,

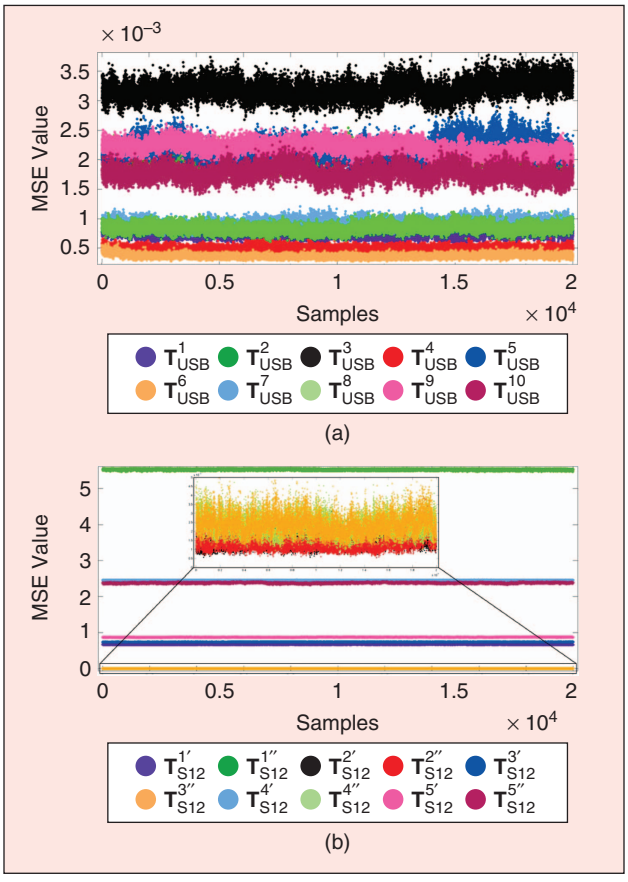

FIGURE 7 The MSE-based separation with (a) a USB-to-CAN device and (b) a Freescale S12 development board for 2×10^4 recorded values (as depicted in [11]).

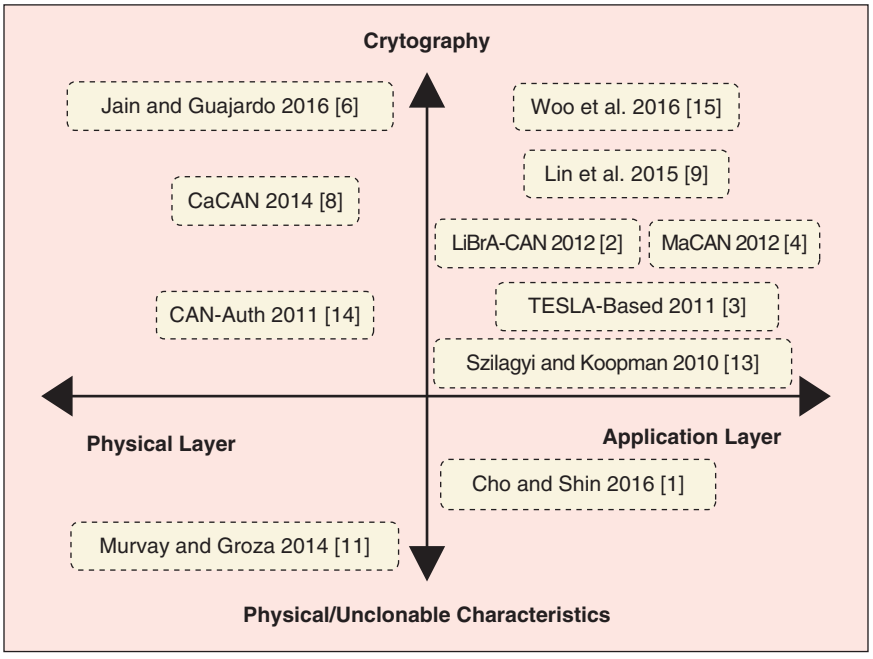

FIGURE 8 The security proposals for CAN, separated by direction toward the physical layer versus the application layer and the cryptographic techniques versus the physical/unclonable characteristics.

physically unclonable functions gained a significant momentum in the security industry. We discuss now the only two lines of work that bring CAN security down to physical characteristics.

Fingerprinting Physical Signals

The use of physical signal characteristics on CAN to distinguish between sender nodes is discussed for the first time in [11]. The main advantage of this procedure is that it does not require cryptography. It thus removes the problem of sharing cryptographic keys or adding bus overheads and is fully backward compatible. The research in [11] shows that one can successfully distinguish between CAN nodes by applying standard mathematical tools on the electrical signal characteristics, e.g., mean-square errors (MSEs) or convolutions. Frames that do not match the expected signal characteristics can be destroyed or signaled with error flags. Figure 7 graphically depicts this separation for transceivers from a universal serial bus (USB) to-CAN device and Freescale S12 development boards based on MSE, as described in [11]. The distance between each frame F_β and the fingerprint Sig_α is computed over each sample of the signal as:

$$MSE(Sig_\alpha, F_\beta) = \sum_{i=1}^{\ell} (Sig_\alpha[i] - F_\beta[i])^2.$$

Experimental results are presented in [11] for two CAN transceivers: PCA82C251 (a high-speed CAN transceiver from USB-to-CAN devices) and TJA1054T (a low-speed CAN transceiver from Freescale S12 development boards). In seven of ten transceivers, identification rates are over 90%. Even with the colliding fingerprints in the other three transceivers, by changing the ID of the message, the overlap rate drops to 0%. This solution may not be easy to implement, but it is the first that does not rely on cryptography for ensuring security on the CAN bus.

Using Clock Skews to Detect Intrusions

Using clock skews is the most recent proposal for detecting intrusions on the CAN bus [1]. This mechanism was successfully explored in the past for source identification in computer networks and mobile phones, but it was never exploited for in-vehicle networks. The main idea is that there are no physically identical oscillators, and variations in the order of several parts per million (ppm) can be used to separate between devices. This is

utilized in [1] by noticing the cyclic nature of the communication on the CAN bus. That is, periodic messages are sent at exact time intervals, e.g., $\delta, 2\delta, 3\delta$, and so on. However, the sender ECU, rather than sending the message from the ith cycle at time $i\delta$, will send it at $i\delta + O_i$, where O_i is the clock offset. Experimental data from [1] shows that clock drifts are separable in the order of dozens of ppm on several real-world vehicles. For example, in the Honda Accord (2013), four sources are determined: 78.4, 199.8, 265.7, and 95.78 ppm. Further difficulties appeared in the Toyota Camry (2010), where two clock skews differed by less than 3%, namely 345 ppm versus 334.1 ppm, which would make these ECUs hard to distinguish. Given the high number of existing ECUs, it seems reasonable to assume that such collisions in the clock drifts may be usual in practice. But the method seems very promising for future investigations.

Discussion and Conclusions

Assuming a similar evolution to that of computer networks, cryptography will become mandatory on in-vehicle buses. The inclusion of cryptographic interfaces in the AUTOSAR standard is a first foundational brick. CAN-FD, the newer embodiment of CAN, offers plenty of space in the frame for adding modern security mechanisms, and there are already many research proposals to be considered for industry adoption.

Figure 8 rounds up the discussed solutions and separates them along the following lines: application layer versus the physical layer and cryptography versus physical characteristics. There is no doubt that cryptographic MACs will stay behind, ensuring authentication; the challenge remains in how they are used, i.e., the concrete authentication protocol. In this respect, group key sharing offers more advantages in front of the basic pairwise key sharing. Given the cyclic nature of communication on in-vehicle buses, TESLA-like protocols should gain more momentum and be of interest for time-triggered networks, e.g., FlexRay. Physical properties of the signal or of existing electronics, e.g., clock drifts, can form the basis of intrusion detection mechanisms.

Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project PN-II-RU-TE-2014-4-1501 (2015–2017).

Author Information

Bogdan Groza (bogdan.groza@aut.upt.ro) earned his Dipl.Ing. and Ph.D. degrees in 2004 and 2008, respectively, from the University of Timisoara (UPT), where he is currently an associate professor. At UPT, he has been actively involved with the development of laboratories by Continental Automotive and Vector Informatik, two world-class manufacturers of automotive software. In 2016, he successfully defended his habilitation thesis on the design of cryptographic security for automotive embedded devices and networks. He currently leads the Cryptographic Security for Automotive Embedded Devices and Networks project, a two-year research program in automotive security.

Pal-Stefan Murvay (pal-stefan.murvay@aut.upt.ro) earned his B.Sc., M.Sc., and Ph.D. degrees in 2008, 2010, and 2014, respectively, from the University of Timisoara (UPT), where he is currently an assistant professor. Prior to joining UPT, he worked as a software developer in the automotive industry for the Continental Corporation from 2005 to 2014. His current research interests include automotive security. He works as a postdoctoral researcher in the Cryptographic Security for Automotive Embedded Devices and Networks project.

References

[1] K.-T. Cho and K. G. Shin. "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Security Symp.*, Austin, TX, Aug. 2016, pp. 911–927.

[2] B. Groza, P.-S. Murvay, A. Van Herrewege, and I. Verbauwhede, "LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks," in *Proc. 11th Int. Conf. Cryptology and Network Security*, Darmstadt, Germany, Dec. 2012, pp 185–200.

[3] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2034–2042, Jan. 2013.

[4] O. Hartkopp, C. Reuber, and R. Schilling, "MaCAN-message authenticated CAN," in *Proc. 10th Int. Conf. Embedded Security Cars (ESCAR)*, 2012.

[5] T. Hoppe and J. Dittman, "Sniffing/replay attacks on CAN buses: a simulated attack on the electric window lift classified using an adapted cert taxonomy," in *Proc. 2nd Workshop on Embedded Systems Security (WESS)*, Salzburg, Austria, Oct. 2007, pp. 1–6.

[6] S. Jain and J. Guajardo, "Physical layer group key agreement for automotive controller area networks," in *Proc. Conf. Cryptographic Hardware and Embedded Systems*, Santa Barbara, CA, Aug. 2016.

[7] 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," in *Proc. IEEE Symp. Security and Privacy*, Berkeley/Oakland, CA, May 2010, pp. 447–462.

[8] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "CaCAN - centralized authentication system in CAN (Controller Area Network)," in *Proc. 14th Int. Conf. Embedded Security Cars (ESCAR)*, 2014.

[9] C.-W. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware modeling and efficient mapping for CAN-based real-time distributed automotive systems," *IEEE Embedded Syst. Lett.*, vol. 7, no. 1, pp. 11–14, Mar. 2015.

[10] C. Miller and C. Valasek, *A Survey of Remote Automotive Attack Surfaces*. Las Vegas, NV: Black Hat USA, 2014.

[11] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 395–399, 2014.

[12] A. Perrig, R. Canetti, J. Tygar, and D. X. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Security and Privacy*, Berkeley, CA, May 2000, pp. 56–73.

[13] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," in *Proc. 5th Workshop Embedded Systems Security*, Scottsdale, AZ, Oct. 2010, p. 10.

[14] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus," in *Proc. 9th Embedded Security Cars Conf.*, 2011.

[15] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle CAN-FD," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2248–2261, Aug. 2016.

VT