CHALMERS
UNIVERSITY OF TECHNOLOGY

In-vehicle CAN message authentication: An evaluation based on industrial criteria

Citation for the original published paper (version of record):
Nowdehi, N., Lautenbach, A., Olovsson, T. (2017). In-vehicle CAN message authentication: An evaluation based on industrial criteria. IEEE Vehicular Technology Conference, 2017-September: 2413-2419. http://dx.doi.org/10.1109/VTCFall.2017.8288327

In-vehicle CAN message authentication: An evaluation based on industrial criteria

Nasser Nowdehi, Aljoscha Lautenbach and Tomas Olovsson
Department of Computer Science and Engineering
Chalmers University of Technology
SE 412 96 Gothenburg, Sweden
{nasser.nowdehi,aljoscha,tomas.olovsson}@chalmers.se

***Abstract*—This is the authors' version of this paper. The final authenticated version is copyrighted by IEEE and is available online at https://www.doi.org/10.1109/VTCFall.2017.8288327.**

Vehicles have evolved from mostly mechanical machines into devices controlled by an internal computer network consisting of more than 100 interconnected Electronic Control Units (ECUs). Moreover, modern vehicles communicate with external devices to enable new features, but these new communication facilities also expose safety-critical functions to security threats. As the most prevalent automotive bus, the Controller Area Network (CAN) bus is a prime target for attacks. Even though the computer security community has proposed several message authentication solutions to alleviate those threats, such solutions have not yet been widely adopted by the automotive industry.

We have identified the most promising CAN message authentication solutions and provide a comprehensive overview of them. In order to investigate the lack of adoption of such solutions, we, together with industry experts, have identified five general requirements they must fulfill in order to be considered viable in industry. Based on those requirements, we analyze and evaluate the identified authentication solutions. We find that none of them meet all the requirements, and that backward compatibility and acceptable overhead are the biggest obstacles.

I. Introduction

Most electrical and mechanical features of modern vehicles are controlled by interconnected Electronic Control Units (ECUs), which, together with sensors and actuators, compose the in-vehicle network. ECUs control many different components of a vehicle, for instance the engine, the windows and the infotainment system. ECUs also control safety features and driver assistance systems such as the anti-lock braking system, adaptive cruise control, the lane keep assistant and the collision warning/avoidance system.

In recent years, the interest in automotive security has increased significantly. In 2015, Miller and Valasek demonstrated how to compromise a Jeep Cherokee [1] remotely, and how to take control of dashboard functions, steering, transmission and brakes. This lead Fiat Chrysler to recall 1.4 million cars [2]. The increased number and effectiveness of such attacks has called for an increased urgency of developing security solutions [3], [4], [5]. While researchers have proposed several solutions for securing in-vehicle networks, few, if any, have been adopted in practice [5], [6]. The introduction of message authentication on Controller Area Network (CAN) buses would strengthen vehicular security considerably, but it also poses big practical challenges.

We make three main contributions: (1) we identify the most promising CAN message authentication solutions and provide a comprehensive overview of them, (2) we identify five requirements for viable security solutions and (3) we evaluate the identified solutions based on those criteria.

II. Methodology

We have identified five requirements which a CAN message authentication solution must fulfill in order to be considered for implementation. This was done together with experts from a passenger vehicle manufacturer. Their expertise included in-vehicle network architecture, in-vehicle network security, automotive Ethernet, CAN, software integrity and remote software updates. We also validated the requirements with security experts from a heavy-duty vehicle manufacturer. These requirements formed the criteria for our analysis of a solution's practical applicability.

We started the literature review with recent surveys of in-vehicle network threats and searched the references for papers on CAN authentication. This was combined with keyword searches in four relevant databases.

To this set of about 30 papers, we applied two simple filters. First, the papers had to be automotive specific. It is possible, for instance, to find papers on low-overhead authentication solutions for Wireless Sensor Networks (WSNs), but while there are similarities to automotive networks, there are characteristics which make the solutions hard to adapt to automotive systems, for instance real-time requirements. Second, we only consider papers which have sufficient technical detail. In the early 2000's, when the field of automotive security was in its infancy, most papers concentrated on defining the problem and only sketched high-level solutions, rather than describing specific solutions to specific problems.

This filtering process left us with ten papers, which, in our judgment, represent the best available solutions in the literature for this particular purpose.

III. The In-Vehicle Network

Vehicular systems use a number of different communication technologies with very specific characteristics. In-vehicle networks usually consist of several domains connected through a backbone, and each domain has one or more buses for different purposes, as shown in Figure 1. Each bus technology

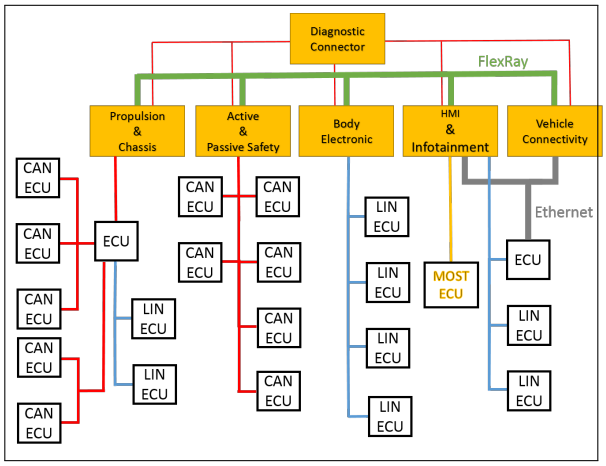

Fig. 1. Typical in-vehicle network with a FlexRay backbone

has particular speed and timing characteristics suited for a specific domain. The most prevalent buses are LIN, MOST, CAN and FlexRay. New technologies are also making their way into automotive systems: Ethernet has already seen its debut and is currently being standardized for use in automotive systems. CAN with Flexible Data-Rate (CAN-FD), which is a backward compatible version of CAN with a higher data rate, is also a candidate for adoption.

The CAN bus is the most common automotive bus: it is well established and well understood, and provides sufficient performance for most applications. A node on a CAN bus can only transmit when it senses the transmission channel to be free. When two or more CAN frames are transmitted simultaneously, the highest priority frame will be transmitted successfully while all other frames will be dropped. Thanks to this prioritization, CAN can be analyzed for its worst-case real-time properties. The typical speed of a CAN bus is 500 kbit/s, with a *maximum speed of 1 Mbit/s*. The main fields of a single CAN frame are the 11-bit ID (or 29-bit in extended format), which is used for arbitration, a control field, which includes the payload size, a data field with a variable payload of 1 - 8 bytes, a Cyclic Redundancy Check (CRC) field and an acknowledgment field. Even though CAN always broadcasts messages, CAN controllers allow to filter unwanted messages. This means that, based on the number of intended receivers, the message can be seen as either unicast or multicast. CAN buses are used for standard operational functions, but also for many safety-critical functions.

The CAN bus has no built-in support for any security measures. All messages are broadcast and it is easy to fake messages, since the identity of the sender can not be verified. CRCs guard only against random transmission errors, so the integrity of the messages can also not be verified if an attacker tampers with the data. The CAN bus has no measures to guard against fraudulently repeated messages. Due to the safety-critical nature of automotive systems, the operations of the vehicle must be protected from manipulation. A message authentication solution ensures that (1) the transmitted data has not been tampered with, (2) the source of the data can be verified and (3) the previously received messages are not being played back by an adversary.

IV. Industrial Requirements for Security Solutions

Over the past two decades, many solutions to various security challenges in the automotive industry have been proposed by security researchers. However, to be able to implement a particular solution in practice, there are a number of requirements which need to be fulfilled.

In collaboration with industry experts, we have identified five industrial requirements (IR) a security solution needs to fulfill in order to be usable in practice. They are:

IR 1: Cost-effectiveness
IR 2: Backward compatibility
IR 3: Support for vehicle repair and maintenance
IR 4: Sufficient implementation details
IR 5: Acceptable overhead

Some of these requirements are absolutely non-negotiable, such as backward compatibility, while others are highly desirable, such as sufficient implementation details.

A. Cost-effectiveness (IR 1)

Cost is one of the main restraining factors for implementing new security solutions. Note that market pressures usually dictate prices in the automotive industry, which means that car manufacturers are bound to particular sales prices, independent of production costs.

Low-cost solutions are needed to stay competitive. For example, requiring all ECUs to have hardware-supported cryptographic primitives to achieve the performance necessary for safety-critical real-time systems is not cost-effective. Strong incentives must be in place for a company to accept the additional cost of adding cryptographic hardware modules.

In order to be cost-effective, an ideal security solution works on existing hardware and under existing constraints. More realistically, a good solution offers an upgrade path for gradual improvement.

B. Backward compatibility (IR 2)

Backward compatibility means that a new solution must be able to co-exist with existing technology and implementations. Backward compatibility is highly desirable, especially because it allows for partial upgrades, i.e., not all ECUs need to be upgraded simultaneously.

Especially for a message authentication solution in heavy-duty vehicles, such as buses and trucks, backward compatibility is a very strong requirement, because the commonly used standard SAE J1939 specifies the entire vehicle communication layer, including specific message content. Consequently, only a message authentication solution which manages the authentication in an extra frame can be considered backward compatible. In order for our results to be widely applicable we use this very stringent interpretation, but it should be noted that SAE J1939 is not used for passenger cars and this requirement can be relaxed for vehicles in that domain.

C. Support for vehicle repair and maintenance (IR 3)

Replacing electronic control units is a regular task in a vehicle's life-cycle. Repair and maintenance may be performed both in authorized and in independent repair shops.

Retrieving data, such as cryptographic keys, from a broken ECU may be impossible, e.g. due to physical damage or hardware failure. Therefore, replacement ECUs must be able to function independently of the data stored on the broken ECU. Consider for instance a solution which stores synchronized message counters locally to guard against replay attacks: a newly added ECU which implements this solution must not depend on retrieving the counter values from the broken ECU to function correctly.

Even if a solution does not explicitly discuss repair and maintenance, we still consider the requirement fulfilled if only minor adjustments to the proposed solution are needed to support it.

D. Sufficient implementation details (IR 4)

Prototyping is a crucial step in demonstrating the viability of a solution before and during the design phase. The lack of a prototype does not mean that a solution is less useful, but prototypes help system designers to evaluate the hardware, software and cost requirements.

Ideally, a solution should have a publicly available prototype, so that it can be evaluated for performance, e.g. memory usage and timing characteristics. More realistically, as few researchers or companies provide prototypes, the solution should at least provide enough detail to enable developers to implement and evaluate it.

A missing implementation is acceptable if a solution is trivial enough to be implemented and evaluated with the given information. If, however, the missing implementation is due to the need for re-designed hardware or a new architecture, which in itself is an indicator of a complex and costly solution, we consider it unacceptable.

E. Acceptable overhead (IR 5)

Automotive networks have strict resource constraints and real-time requirements, which in turn impose strict constraints upon the security solutions. The real-time requirements are often due to safety considerations: when the brake pedal is engaged, the reaction must be near instantaneous (less than 50 ms). In other cases, the real-time requirements are simply the result of user expectations: when pressing the door unlock button, we expect the reaction to be instantaneous.

Regarding the proposed security solutions, we are interested in the overhead they impose on the CPU, the message latencies, the bus load and the storage. It is important to note that during the evaluation of the solutions, we consider the currently available technology in the vehicles as baseline for acceptable overhead. If a solution assumes additional hardware to achieve the necessary performance, it does not have acceptable overhead according to our criteria.

Where appropriate, we make use of the results by Vasile et al. [6] for the discussion of acceptable overhead. They conducted performance evaluations for four of the solutions we analyze, and observed patterns in performance for different kinds of key distributions.

TABLE I
EVALUATED CAN AUTHENTICATION SOLUTIONS

Name	Paper	Year
CANAuth	Van Herrewege et al. [7]	2011
SchweppeAuth	Schweppe et al. [8]	2011
LiBrA-CAN	Groza et al. [9]	2012
LinAuth	Lin and Sangiovanni-Vincentelli [10]	2012
MaCAN	Hartkopp et al. [11]	2012
CaCAN	Kurachi et al. [12]	2014
VeCure	Wang and Sawhney [13]	2014
WooAuth	Woo et al. [14]	2014
VatiCAN	Nürnberger and Rossow [15]	2016
WeisglassAuth	Weisglass and Oren [16]	2016

V. Description and Evaluation of Message Authentication Solutions

In this section we outline the proposed security solutions and analyze them according to the five requirements introduced in section IV. Table I lists the analyzed message authentication solutions and their names (first author + "Auth" if name is unknown). A summary of the results can be found in Table II: a check mark (✓) indicates that the requirement is fulfilled, a question mark (?) indicates that it is difficult to judge whether a solution fulfills the requirement, and an X (✗) indicates that the requirement is not fulfilled.

Even though the evaluation of the provided level of security of the solutions is not the purpose of this paper, we provide a rough estimate by assigning three approximate security levels: weak, medium and strong. The intention is to give a better overview of the solutions overall utility. *A detailed security analysis of the solutions is out of the scope of this paper.*

A. CANAuth

Van Herrewege et al. [7] propose CANAuth which is intended to work on CAN+ [17], a theoretical extension of CAN which increases the typical CAN speed up to 16 times. To date, no implementation of CAN+ exists. CANAuth employs 128-bit pre-shared group keys for establishing group session keys of the same length, and every authenticated message adds a 80-bit HMAC and a 32-bit counter to guarantee the message freshness. Nodes store the counter of the last valid message and the session key needs to be renewed every time the counter overflows. Note that they propose to use one key for every message ID, which requires a large number of keys to be stored on every ECU, one for every type of message it sends and receives.

Cost-effectiveness (✗) CAN+ currently does not exist in practice and it is costly to implement it mainly because it requires new CAN controllers to be designed.

Backward compatibility (✗) CANAuth is intended to work on CAN+ and is thus backward-incompatible.

Repair and maintenance (✗) Since there is no central key storage unit, it is unclear how ECU replacement would work.

TABLE II
EVALUATION OF THE MESSAGE AUTHENTICATION SOLUTIONS ACCORDING TO THE IDENTIFIED REQUIREMENTS.

Message Authentication Solution	IR 1 Cost Effectiveness	IR 2 Backward Compatibility	IR 3 Repair and Maintenance	IR 4 Implementation Details	IR 5 Acceptable Overhead	Approx. Security Level[1]
CANAuth [7]	✗	✗	✗	✗	✗	Strong
SchweppeAuth [8]	✗	✗	✓	✓	✗	Strong
LiBrA-CAN [9]	✓	✗	✓	✓	✗	Strong
LinAuth [10]	?	?	?	✗	?	Medium
MaCAN [11]	✓	✓	✓	✗	✗	Medium
CaCAN [12]	✓	✗	✓	✓	?	Weak
VeCure [13]	?	✓	?	✓	?	Medium
WooAuth [14]	✓	✗	✓	✓	✓	Medium
VatiCAN [15]	✓	✓	✓	✓	?	Medium
WeisglassAuth [16]	✓	✗	✗	✓	?	Medium

Sufficient implementation details (✗) Since CAN+ is a theoretical construct, no implementation was done.

Acceptable overhead (✗) Van Herrewege et al. only provide a theoretical analysis of the minimum transmission times for CANAuth, which makes it difficult to judge the actual performance of their solution.

B. SchweppeAuth

Schweppe et al. [8] propose a group session authentication protocol within a larger framework which includes entity authentication and policy-based access control. They assume that each ECU has a hardware security module (HSM) to securely store keys, to accelerate cryptographic primitives, and to enforce that each key is used for only one particular purpose, thus enabling asymmetric use of symmetric keys. They use one or multiple Key Masters (KM) to control the key distribution process. Each ECU uses two pre-shared keys with the KM, one for authenticating itself, and one for transporting generated session keys. The session keys must be generated by the sender ECU and exchanged with the group of receivers via the KM using the transport keys. However, due to the asymmetric use of the keys, every ECU in a group requires one sending key, and one receiving key for every other group member. The authors also propose to use Message Authentication Code (MAC) truncation with a minimum length of 32 bits. Schweppe et al. point to a technical report for more details, in which it is described that a global time is maintained by the HSM, and timestamps can be added to the HMACs on demand.

Cost-effectiveness (✗) It is cost-ineffective as every ECU which implements this solution needs an HSM module.

Backward compatibility (✗) Schweppe et al. propose to use an adapted transport protocol to circumvent the message size restriction, which according to our criteria is not backward compatible.

Repair and maintenance (✓) Even though Schweppe et al. do not discuss ECU replacements, their scheme can be adapted so the key master can be queried for particular ECU keys by authorized tools.

Sufficient implementation details (✓) The authors describe their implementation in sufficient detail.

Acceptable overhead (✗) The experiments by Vasile et al. [6] suggest that for solutions with pairwise keying, such as this one, the additional bus load can not be handled on CAN.

C. LiBrA-CAN

Groza et al. propose LiBrA-CAN [9], an authentication protocol based on key splitting and MAC mixing. They introduce several variants which all share the same characteristics: they introduce rather complex key management, and for every data frame that must be authenticated, several authentication frames are required. As a result, LiBrA-CAN has very high bandwidth requirements.

Cost-effectiveness (✓) No additional hardware is required, so it is cost-effective.

Backward compatibility (✗) In the proposed form, LiBrA-CAN is not backward compatible according to our criteria since it requires to set specific message IDs for the data and authentication frames.

Repair and maintenance (✓) Even though Groza et al. do not discuss ECU replacements, their scheme can be adapted so the key master can be queried for particular ECU keys by authorized tools.

Sufficient implementation details (✓) They provide sufficient implementation details.

Acceptable overhead (✗) They evaluated the performance of their protocol on real hardware with both low and high-end ECUs. The bandwidth requirements are prohibitive for regular CAN, so the overhead is unacceptable.

D. LinAuth

Lin and Sangiovanni-Vincentelli [10] propose a simple MAC authentication scheme using pre-distributed pair-wise keys. In case of multiple receivers of a message, they propose that the MACs for each receiver be concatenated in the message, and each receiving ECU only verifies the MAC addressed to them. Counters for each message ID are stored locally to guard against replay attacks, and the acceptance criteria is that the received counter must be larger than the previously stored counter. Since this requires large counters to avoid overruns, they propose to only send the least-significant bits of the counter, while still

[1]The security level is not part of the evaluation criteria

using the entire counter internally for the MAC. Unfortunately Lin and Sangiovanni-Vincentelli are extremely vague on the algorithmic part of the MAC calculation. They provide no algorithm recommendation and no discussion of the MAC length or possible splitting to fit the MAC into a CAN frame.

Cost-effectiveness (?) They propose a software-only solution, but without more implementation details, it is hard to judge if it is cost-effective.

Backward compatibility (?) It is not specified how the authentication frames are to be sent, which makes it impossible to judge whether this solution is backward compatible.

Repair and maintenance (?) ECU replacements are not considered in this scheme.

Sufficient implementation details (✗) Their results can not be replicated due to lack of detail regarding the MAC.
Acceptable overhead (?) They assume a realistic 75% bus utilization, but without additional information, the overhead can not be properly evaluated.

E. MaCAN

Hartkopp et al. [11] devised a scheme for on-demand authentication using a 32-bit Cipher-based Message Authentication Code (CMAC). It requires a key server for long-term and session key management and a time server to protect the messages against replay attacks. MaCAN proposes to group ECUs based on their trust level to share the same symmetric key. The authors suggest that the trust level can either be set by the ECU manufacturer or be determined by the available level of hardware security. Bruni et al. [18] identified, and proposed improvements for, two flaws in the original specification of MaCAN.

Cost-effectiveness (✓) MaCAN is cost-effective because the time server can be implemented on an existing ECU and no additional hardware support is required.

Backward compatibility (✓) MaCAN is backward compatible as it optionally allows to manage authentication in an extra CAN frame and it leaves the original frame intact.

Repair and maintenance (✓) Even though the authors do not specify any procedure for ECU replacement, their solution can be adapted for retrieving particular keys from the key master by authorized tools.

Sufficient implementation details (✗) Despite describing MaCAN in detail, Hartkopp et al. provide no performance information from their simulations.

Acceptable overhead (✗) The performance analysis done by Vasile et al. [6] shows that the overhead of MaCAN is prohibitive if the number of groups is too high.

F. CaCAN

Kurachi et al. [12] propose a centralized architecture in which every CAN bus, which requires authentication includes a powerful monitoring node. On such a bus, every ECU attaches an 8-bit truncated MAC and an 8-bit truncated counter to its messages, but only the monitoring node verifies the signature. When authentication fails, the monitoring node can discard messages by overriding them with an error frame, using a special CAN controller. However, the monitoring node is a single point of failure: if it is compromised or fails, the entire system's security is compromised.

Cost-effectiveness (✓) CaCAN is cost-effective because the only required addition is a single monitoring node.

Backward compatibility (✗) Since CaCAN alters the CAN frame payload by reserving its last 2 bytes for the counter and MAC, it is backward incompatible.

Repair and maintenance (✓) Even though CaCAN does not explicitly consider ECU replacement in its design, it has a straight-forward key management system which facilitates the repair and maintenance of the ECUs.

Sufficient implementation details (✓) They provide sufficient implementation details.

Acceptable overhead (?) Vasile et al. [6] assume that a single key is used with CaCAN, and under that assumption it offers good performance and rather weak protection. The problem is that every authenticated message needs 2 bytes in the existing frame, which may create a significant additional bus load if there are many frames with more than 6 bytes of data. Therefore, the overhead is hard to judge.

G. VeCure

Wang and Sawhney [13] group ECUs into low-trust, medium-trust, and high-trust groups, based on their exposure to external interfaces. ECUs in the high-trust group share two symmetric keys, one for the high-trust group, and one for the medium-trust group. ECUs in the medium-trust group share only the key for their own group. The authentication function has two parts: a light-weight online calculation component and a heavy-weight offline calculation component. For every message which needs to be authenticated, an extra authentication message is sent. This solution mitigates the replay attacks by using a 2-byte session number that is incremented by 1 each time the vehicle starts, and a 2-byte message counter for each message ID. The message counters must be maintained by both the sender and the receiver and transmitted in the authentication message.

Cost-effectiveness (?) To prevent replay attacks, every ECU maintains session-bound message counters for every particular message ID for all ECUs it communicates with. Consequently, the number of message counters an ECU needs to maintain can be significant in a larger network. In such a case, adopting the solution may require ECUs with more storage. Therefore, it is difficult to judge whether the solution is cost-effective.

Backward compatibility (✓) VeCure is backward-compatible because it handles the authentication in an extra CAN frame. This solution can be implemented for a small group of ECUs in the beginning and be gradually introduced to the network.

Repair and maintenance (?) VeCure stores session IDs in flash memory on every ECU and retrieves them each time the vehicle starts, which makes the repair and maintenance of broken ECUs difficult, if not impossible.

Sufficient implementation details (✓) The authors provide enough implementation details.

Acceptable overhead (?) Wang and Sawhney only evaluate the processing delay of VeCure and they do not provide information about its bandwidth overhead.

H. WooAuth

Woo et al. [14] propose that CAN data frames should be encrypted and authenticated with AES-128 and HMACs. WooAuth also requires each ECU to reject messages if the sender is not registered and to store a message counter for each sender. On start-up, all ECUs contact their gateway ECUs to run the Authenticated Key Exchange Protocol 2 (AKEP2) [19] to derive initial authentication and encryption keys. The keys must be updated for each session. The protocol adapts CAN's extended ID and CRC fields to carry 32 bit authentication codes. They assume that each ECU loads and stores long-term symmetric keys created during manufacturing, and updated when ECUs are replaced.

Cost-effectiveness (✓) WooAuth is cost-effective. It does not require additional hardware and it only assumes that the gateway ECUs are more powerful than the regular ECUs, which is an acceptable assumption.

Backward compatibility (✗) WooAuth alters the CAN frame ID and CRC fields to store authentication data, and is thus backward incompatible.

Repair and maintenance (✓) Woo et al. specify procedures for sharing and updating keys with diagnostic tools which are necessary for repair and maintenance.

Sufficient implementation details (✓) WooAuth has been implemented and its procedures are described in detail.

Acceptable overhead (✓) They performed a mix of hard- and software based simulations. WooAuth has acceptable run-time and bus overhead on a CAN bus that has up to 20 regular vehicular ECUs and a powerful gateway.

I. VatiCAN

Nürnberger and Rossow [15] propose VatiCAN, in which CAN data frames should be authenticated with SHA-3 HMACs in separate authentication frames. The authentication frame j for message i is assigned the ID $i + 1$, i.e., its priority is just below the original message. This avoids possible priority-inversion scenarios which could happen due to the delayed message verification. Nürnberger and Rossow also propose local message counters of 4 bytes to guarantee freshness. The counters are (re-)synchronized by a global Nonce Generator (NG), which periodically broadcasts a nonce which must be adopted by all local counters. They suggest a period of 50 ms for the NG, since this represents the maximum time an ECU may fail to authenticate messages when the counters get out of sync. Every ECU receives its own symmetric keys during production, which means that every ECU must store the key of every other ECU it exchanges authenticated messages with.

Cost-effectiveness (✓) VatiCAN only needs an additional global Nonce Generator and a small amount of additional storage, so it is cost-effective.

Backward compatibility (✓) Since extra authentication frames with new message IDs were chosen, VatiCAN provides full backward compatibility. A caveat is that no procedure to determine a message priority is given, in case ID $i + 1$ is already taken.

Repair and maintenance (✓) ECU replacements require changing the symmetric key for the ECU being replaced on all ECUs which communicate security with it. This also requires the vehicle manufacturer to have a central key database which can be accessed by third-party workshops.

Sufficient implementation details (✓) Nürnberger and Rossow claim to have implemented an open source library for VatiCAN, which they tested on a low-end ECU in a small, realistic setup. Unfortunately, at the time of writing, the library was not available.

Acceptable overhead (?) Their tests reveal that total processing time for an authenticated message is about four times slower than an unauthenticated message. As the authors note this may be acceptable for a low number of high priority messages, but it would not scale for an entire subsystem.

J. WeisglassAuth

Weisglass and Oren [16] propose a solution based on "lightweight cryptography" that achieves authentication through encryption of CAN data. They use the End-to-End (E2E) protection portion of safety related messages for authentication. The End-to-End protection mechanism usually includes a CRC and a sequence counter for error-detection. Their solution can be used for E2E safety protected links only, which implies there is only one recipient for a message. Message integrity is assured using a CRC value that is generated from the concatenation of the counter and the payload of the CAN message. In the proposed scheme, the CAN data is encrypted by the sender and decrypted by the receiver using a shared key. After decryption, the receiver must check the retrieved CRC value to verify the integrity of the message. Weisglass and Oren propose that when a CAN message is sent to ECUs that do not support encryption, the message must be sent twice, once in clear-text and once as encrypted data. Therefore, the proposed solution can also be used on multicast links but at the expense of adding network overhead. WeisglassAuth is protected against replay attacks by using the combination of E2E counter and periodic key rollovers. They discuss key management issues, but they give no recommendation which mechanism to use.

Cost-effectiveness (✓) The solution is cost-effective since it works on existing hardware and does not require additional hardware support for implementing its lightweight cryptographic algorithms.

Backward compatibility (✗) Even though the solution considers handling authentication in an extra frame for ECUs that do not support this solution, it is backward incompatible because it alters the CAN frame payload to authenticate messages on unicast links.

Repair and maintenance (✗) Although the solution considers repair and maintenance as a design criterion, it is not practical to use if the diagnostic tools at independent repair shops do not support the decryption.

Sufficient implementation details (✓) Weisglass and Oren provide enough details about their implementation.

Acceptable overhead (?) Even though WeisglassAuth performs very well with respect to run-time and bus load, the absence of detail about the evaluation setup preclude us from drawing any conclusion about its actual performance.

VI. Evaluation Synthesis

Several interesting observations can be made from Table II. Most solutions are cost-effective, if we assume that we only implement them on a small subset of safety-critical ECUs. On the other hand, only three of the solutions can meet our rather strict interpretation of backward compatibility. For a less strict interpretation, several more could be deemed backward compatible. While support for repair and maintenance is rarely considered explicitly, in most cases it can be addressed without undue effort. Regarding sufficient implementation details, more than half of the solutions provide enough details to properly evaluate their performance and to be able to implement the solution in real life. Finally, about half of the solutions have an unacceptable overhead, and for the other half it is not possible to judge because the evaluation environment is not well explained, with the notable exception of WooAuth. Most evaluations point to the fact that the constraining factor for CAN authentication is the bus load. The run-time overhead is significant, but in most cases it is not a showstopper.

Two of the most recent candidate solutions might be worth considering for industry adoption, each for a different use-case. WooAuth [14] meets almost all criteria, but unfortunately it is not backward compatible. Also, using a large part of the ID to transmit the MAC severely limits the available message IDs per bus. It might be well-suited for new designs which do not require backward compatibility.

VatiCAN [15] on the other hand was designed with full backward compatibility in mind, and Nürnberger and Rossow consider many practical design issues. However, if VatiCAN meets the performance criteria is unclear. Therefore it could be considered for adoption in current designs.

This highlights the difficulty to fulfill the requirements for backward compatibility and acceptable overhead at the same time, due to the restrictions of the CAN bus.

VII. Conclusion

In recent years, many solutions have been proposed for securing in-vehicle communications, but there is still a lack of practically feasible solutions. We identified five general requirements which a security solution needs to fulfill in order to be considered for adoption. We reviewed ten CAN bus message authentication solutions proposed in the literature, and evaluated them according to the identified requirements.

The evaluation shows that none of the proposed CAN authentication solutions meets all of the criteria. Due to the tight resource restrictions of automotive systems in general, and the CAN bus in particular, meeting the backward compatibility and acceptable overhead expectations are the biggest adoption hurdles. These findings suggest that the CAN bus might be fundamentally unsuited for secure communication, and that a gradual shift towards more modern bus technologies with higher bandwidth is needed, in order to secure internal vehicular communications.

References

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.

[2] *Alert (ICS-ALERT-15-203-01), FCA Uconnect Vulnerability*, July 2015 (Accessed 2-April-2017), https://ics-cert.us-cert.gov/alerts/ICS-ALERT-15-203-01.

[3] 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 *Security and Privacy (SP), IEEE Symposium on*, 2010.

[4] 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 Security Symposium*, San Francisco, CA, USA, 2011.

[5] P. Carsten, T. R. Andel, M. Yampolskiy, and J. T. McDonald, "In-vehicle networks: Attacks, vulnerabilities, and proposed solutions," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. ACM, 2015.

[6] P. Vasile, B. Groza, and S. Murvay, "Performance analysis of broadcast authentication protocols on CAN-FD and FlexRay," in *Proceedings of the WESS'15: Workshop on Embedded Systems Security*, ser. WESS'15. New York, NY, USA: ACM, 2015.

[7] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth - a simple, backward compatible broadcast authentication protocol for CAN bus," in *ECRYPT Workshop on Lightweight Cryptography*, 2011.

[8] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "Car2x communication: Securing the last meter - a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography," in *Vehicular Technology Conference (VTC Fall), IEEE*, 2011.

[9] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede, "LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks," in *Cryptology and Network Security*. Springer, 2012.

[10] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the Controller Area Network (CAN) communication protocol," in *Cyber Security (CyberSecurity), International Conference on*. IEEE, 2012.

[11] O. Hartkopp, C. Reuber, and R. Schilling, "MaCAN - Message authenticated CAN," in *Escar Conference, Berlin, Germany*, 2012.

[12] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "CaCAN - Centralized authentication system in CAN (controller area network)," in *14th Int. Conf. on Embedded Security in Cars*, 2014.

[13] Q. Wang and S. Sawhney, "VeCure: A practical security framework to protect the CAN bus of vehicles," in *Internet of Things (IOT), International Conference on the*, 2014.

[14] S. Woo, H. J. Jo, and D. H. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[15] S. Nürnberger and C. Rossow, "– vatiCAN – vetted, authenticated CAN bus," in *Cryptographic Hardware and Embedded Systems: 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[16] Y. Weisglass and Y. Oren, "Authentication method for CAN messages." ESCAR Europe, 2016.

[17] T. Ziermann, S. Wildermann, and J. Teich, "CAN+: A new backward-compatible controller area network (CAN) protocol with up to 16x higher data rates." in *Design, Automation Test in Europe Conference Exhibition*, 2009.

[18] A. Bruni, M. Sojka, F. Nielson, and H. R. Nielson, "Formal Security Analysis of the MaCAN Protocol," in *Integrated Formal Methods*. Springer, 2014.

[19] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Annual International Cryptology Conference*. Springer, 1993.