A Survey over Low-Level Security Issues in Heavy Duty Vehicles

Luca Dariz, Massimiliano Ruggeri
IMAMOTER-CNR
Via Canal Bianco, 28, Ferrara, Italy
{l.dariz,m.ruggeri}@imamoter.cnr.it

Gianpiero Costantino, Fabio Martinelli
IIT-CNR
Via G. Moruzzi, 1, Pisa, Italy
{gianpiero.costantino,fabio.martinelli}@iit.cnr.it

***Abstract*—This is the era of open architectures, fleet management systems, remote support and other advanced features that can mean added value against competitors in heavy-duty vehicles industry. The need for security measures has emerged for In-Vehicle networks to avoid and prevent a number of issues, which become relevant with the increase of connectivity of such networks. A deep knowledge of the security problems, methodologies and possible solutions has been developed in the IT field, but there are significant differences in the application requirements and the network itself, when it comes to heavy-duty machines, specifically time-triggered communication and network throughput. Although researchers are overtaking network constrains with the recent research on the use of high-speed networks such as Ethernet instead of the classic CAN bus, security issues still remain a major constraint. The basic issues can be identified with an appropriate attacker model, which will be used to identify some security threats, proposing existing solutions and the challenges that they pose. In particular, in this paper we propose solutions for specific security threats, and compare their outcome both for the low-speed CAN and high-speed Ethernet networks, with emphasis on the authentication, integrity and encryption solutions.**

I. Introduction

The increasing interest of Cyber-Physical Systems *(CPS)*, in particular in the automotive field, has exposed new security issues that tackle this kind of vehicles. Cars and heavy-duty machines are not anymore just isolated entities, but nowadays they can be considered devices, like computers, tablets and smartphones, connected to the Internet. This great opportunity gives to cars and heavy-duty machines new important functionalities that makes them more appealing. However, the Internet connectivity make vehicles exposed to cyber-security threats. For instance, malicious users may remotely find and exploit cars vulnerabilities to access the car and perform malicious remote actions. A well-known attack is that on the Jeep Cherokee performed by Valasek and Miller [1] in the summer of 2015. The author of this attacks demonstrated a hack to remotely control a Jeep Cherokee. Basically, Valasek and Miller were able to remotely change the direction of the steering and also to command the breaking system. This attack was performed by communicating with the car through the infotainment system. So, the Fiat Chrysler was forced to provide a software update [2] to be installed through the USB port on their vehicle's dashboard.

Many examples of the attack possibilities and their feasibility are present in literature; for example, in [3] the authors exploit local access to the CAN network to gain complete control of a car, while in [4] they explore the possibilities of a remote attack. These examples, although focused on the automotive field, are still relevant for heavy-duty vehicles, since the increased connectivity that is permeating also this field exposes to similar scenarios, although with different consequence, given the different nature of the vehicles.

Seen the above security risks, our opinion is that past communications protocols and new deployed ones, should be adapted to cover some security requirements and to avoid, or at least mitigate, the security threats. Old protocols may be extended, if possible, to embed security solutions, while new ones should be designed with the Security by Design principles, in which engineers should have in mind the attack surfaces [5] to provide the right security solutions.

In this manuscript we mainly focus on traditional heavy-duty machines that are equipped with one or more CAN-based networks, used by all the electronic control units (ECUs) to communicate. In the last years, due to the increasing demand for bandwidth from the applications implemented in the ECUs, the Ethernet network has been considered as a long-term replacement for the in-vehicle networks in heavy-duty machines. On the other hand, there is also an increasing need for secure communication, mainly due to the fact that in-vehicle networks are now connected directly or indirectly with other networks, and in many cases with the Internet. This increase in connectivity opens new technological and business possibilities but, as reported in [6], it creates a situation where a vulnerability can have disastrous consequence, both in terms of security and safety. The effective implementation of these two scenarios depends to a large degree on the capabilities of the underlying network; as shown in this paper, the traditional CAN networks ultimately impose some limits on the security level that can be obtained in both scenarios, due to the very short maximum packet size. On the other hand, an Ethernet network permits a greater packet sizes, and the security level achievable is basically given by the computational power available on the involved ECU.

In this paper we analyze the current CAN-based networks from three different security properties, which are: *Authetication, Integrity* and *Confidentiality*. In particular, we

consider some relevant and well-known security threats, and we investigate how sound the CAN networks are against the chosen security threats. Moreover, we propose some security solutions, when possible, that could make the current CAN standard more robust. Finally, we make a comparison of the security solutions and threats whether the CAN bus is replaced with the Ethernet network.

In §II we present the attacker model and an overview of the major security threats with emphasis on low-level aspects. §III states the requirements imposed commonly by real-time applications on the network protocol stack. §IV and §V analyze how the security threats can be solved, first analyzing the present-day network (the CAN bus) and then a possible future evolution, based on IEEE 802.1 Ethernet. §VI discusses the impact of the presented solutions, and points to future directions and future works. Finally, §VII concludes the paper.

II. General Security Considerations

In this section we introduce the following security aspects: *attacker model*, *security threats* and *challenges*.

A. Attacker Model

Nowadays, the evolution of communication technology into heavy-duty machines may introduce new vulnerabilities that impact on the proper machine functionalities. In §II-B we list some security threats that attackers could exploit being located far away from the vehicle, or in proximity, using a physical connection with the vehicle, e.g., a USB-port. The choice to consider versatile attackers is enforced by the fact that vehicles are not isolated entities anymore, but they are connected to the Internet through, for instance, 3G/4G or Wi-Fi connectivity. Thus, by exploiting wrong security configuration or security flaws in the protocols, attackers may easily achieve access to the machine without particular effort. In fact, if we consider that a generic communication protocol, developed in the past and used for decades, was not thought to be secure by design, for example without an authentication system, once it is moved into a different domain, e.g., the Internet, attackers can access the vehicle's local network without any hindrance.

B. Security Threats

The lack of proper protection for ECUs inside heavy-duty machines makes them exposed to potential attacks. In particular, the risk of attacks increases when vehicles are connected to the Internet to provide additional services. An attacker may exploit remote connectivity to get access to the machines and execute malicious actions. However, remote attacks usually require considerable effort to be exploited. For instance, if ECUs are configured to request basic authentication, i.e., username and password authentication, to access their internal services, then the attacker must be able to break the authentication system to get partial or full control of the vehicle itself and execute several actions, e.g., commanding the vehicle trajectories, sniffing messages of the ECUs and so on.

In the following we list attacks, well-known into the security field, that can bring relevant security issues to heavy-duty machines. Also, given the nature of the attacks, we classify them into two categories: *Local* and *Remote*. In the first category, an attacker execute the attack being placed in proximity of the target, instead in *Remote*, an attacker exploits the Internet connection to access the vehicle.

Starting with the *Sniffing* attack (*Local*) a malicious user may be able to read the content of all messages exchanged through the CAN bus. In fact, as soon as messages do not contain the confidentiality property[1], any message is vulnerable to this attack.

An attack linked to the sniffing, and in the same local category, is the *Replay* attack. This typology of attack is bounded to the first one since it uses messages captured with the sniffing attack that, then are retransmitted on the network for specific purposes. For example, an attacker may sniff and store communications between the ECU that controls the wheel, and the ECU that controls the wheel movements, to replay the actions without control of the driver.

Moving to the remote attacks, we first describe the *Denial of Service* (DoS) attack. Here, the attacker wants to make unavailable one or more services of the vehicle. This kind of attack is common on the traditional networks, and in particular it involves attacks on Web Servers and Web Services, to make them not reachable from end-users. Within heavy-duty machines an attacker, who performs the DoS attack, can make specific ECUs unavailable , e.g., attacking that one to control the wheels the DoS attack strongly impacts the *safety* of vehicles.

Another remote attack is the *Black hole* [7], which can be considered under the typology of the DoS attacks. In this case, the attacker remotely gets control of one or more ECUs to block and drop messages that transit through them.

C. Security Challenges

As we presented in §II-B, heavy-duty machines may be exposed to security threats that could impact their security and safety. This risk increases when vehicles are connected to the Internet and they become remotely reachable, for instance exploiting the IP address. So, vehicles cannot be seen as isolate entities protected from computer security attacks.

Our opinion is that heavy-duty machines connected to the Internet should be protected from attacks listed above. However, presenting solutions already available "in the market" are not always feasible due to the current physical infrastructure available in the heavy-duty machines. For instance, confidentiality may be considered by importing cryptographic solutions to encrypt messages. Other solutions may mitigate the risk of DoS attacks by using firewalls at the border of the vehicles' network, to block messages flooding that may make services unavailable. Also, message validation techniques may help decreasing the impact of a DoS attack to ECUs by sharing

[1]Generically speaking, for *confidentiality* is meant the possibility to exchange a message between two users without that a third user is able to read the content.

a secret that they will use to evaluate the validity of messages. So, an attacker must be able to forge a valid message to be accepted by the counterpart, and this is not trivial since it requires that the attacker guesses the secret, e.g., an attacker may execute the *Random message* attack to forge random messages.

The above solutions can help the heavy-duty machines to "raise the bar" on their level of protection against security attacks. However, those solutions require a suitable network infrastructure and enough computational resources to be implemented. In this paper, our goal is to analyze the current network infrastructure available in the heavy-duty machines, i.e., CAN bus, to propose solutions that reduce the risk of attacks presented in §II-B. Finally, we compare the same solutions with the Ethernet protocol, and we analyze their feasibility.

III. General Network Considerations

In the following sections, the requirements of typical real-time traffic are considered. For a security measure or algorithm to be applicable, the following constraints must be fulfilled:

1) *Limited computational complexity* - that is, the execution time of the algorithm must be negligible compared with the time repetition of the messages. Ideally this should also be fixed to avoid timing attacks and satisfy real-time requirements;
2) *Minimum overhead on the data packets* - that is, the algorithm overhead in terms of bytes added to the message must be negligible with respect to the network packet constraints, in order to be transmitted without fragmentation. This is required for many real-time messages, where fragmentation is often not tolerable.

While the computational power available on a typical ECU can be expected to increase over time, due to technological advances, the network packet constraints are generally fixed and depending on the network technology used. Furthermore, the network usually also imposes some requirements on the possible topology and overall achievable security level.

IV. Present - CAN network

The CAN bus is one of the most widespread vehicular networks, as is found on automotive, as well as heavy-duty, agriculture vehicles. For a plain CAN network the maximum message size is *8 bytes*; although various Transport Protocols like ISO-TP [8], SAEJ1939 Connection Mode [9] or Extended Transport Protocol [10] are available, which assure larger messages to be transferred, they impose an additional overhead that makes them typically not suitable for real-time traffic, since they split a single block of data over different messages and use explicit acknowledge packets for reliability. A CAN network imposes a network architecture that is a physical shared bus, therefore everyone can listen to the messages generated from other ECUs and can also transmit messages which can be received by any ECU. This exposes to *Replay attacks* and data collection (*Sniffing attack*) for offline attacks on the algorithms used, as well as *DoS* and *Black Hole* attacks.

A typical vehicular network is composed of more than one CAN subnetwork, usually identified by the main purpose of the connected ECUs, e.g. the powertrain network, the implement (e.g. ISOBUS) network. A single ECU may have more than one CAN interface, and can be connected to more than one CAN subnetwork. To mitigate these attacks, CAN gateways could be employed, but in practice they are rarely used.

The different CAN subnetwork can also have different exposures; usually the powertrain network is an internal network, while the ISOBUS network is open to the connection of implements, which can be out of the control of the manufacturer. Similarly, there is usually a diagnostic network, which can be separated from the others but can also be part of an existing CAN subnetwork. It must be noted that currently mulltiple networks are generally adopted not to increase security, but because a single network is not sufficient for the throughput requirements of the applications, or because of some dedicated sensor network.

A. Authentication over CAN bus

Currently, on a CAN network, authentication is only performed to enable certain functionalities, typically for mainteinance purposes. The protocol used is generally called *Seed and Key*, and is used for example in the CAN Calibration Protocol (CCP) [11] and in the Keyword 2000 protocol (KW2000) [12]. This protocol is performed in two phases:

1) the client ECU that wishes to authenticate on a server ECU requests a seed S. The client ECU then computes the key K, according to a certain algorithm, depending on S and optionally on a fixed shared key k_s between the client and the server;
2) the server ECU receives K from the client ECU, and verifies its correctness in order to decide if the functionality has to be enabled, usually by recalculating it with the same algorithm used by the client ECU.

Neither the CCP nor the KW2000 protocols specify the algorithm to be used for the *Seed and key*; such algorithm can depend on the application, the manufacturer and the functionality. These algorithms usually rely on security through obscurity (for example $K = f(S, k_s)$ where f is a secret function, which sometimes is a simple addition, or a bitwise xor). It is worth noting that the main purpose of the *Seed and key* protocol is to reduce the chance of unauthorized operations; however sometimes the secret algorithm is discovered, by means of reverse engineering, for example for tuning or cloning of ECUs. The *Seed and key* protocol is usually vulnerable to the *Replay attack*, since for a given seed S_i there is only one possible key K_i. Although the manufacturer of an ECU can limit the number of *Seed and key* transactions per second to mitigate the possibility of a *Replay attack*, the number of S_i, K_i pairs that can be obtained in reasonable time is normally sufficient for reverse engineering the secret algorithm. The *Seed and key* protocol as currently used is clearly not sufficient for the purposes of this paper, since here the data must be continuously authenticated, and the authentication must be difficult to break. This means that every

CAN message should be authenticated, for example using a Message Authentication Code (MAC). The authentication of a CAN message, whose size is limited to 8 bytes, can be problematic since it requires an additional field to be added to the message. Using multiple messages, is not a feasible solution for real-time traffic, as noted in section III. While it could be possible to limit the data message length reserving a number of bits for a MAC, another solution could be to use the existing 15 bit normally used for the CRC field to store a 15-bit MAC code, therefore substituting to the integrity check, or to allow the use of DLC in the range 9-15. This has both the disadvantage of a limited MAC length and non-standard behavior, but allows for full 64 bit messages to be authenticated. However the need for a custom transceiver is likely to be a major hindrance for these extension of the CAN protocol. A similar approach with a modified CAN bus is presented in [13].

Given the size constraint, the strength of a MAC depends not only on the algorithm used, but also on the length of the MAC itself; in a standard CAN message, this will be the number of unused bytes. Using the CMAC scheme, an evaluation of the minimum MAC length with regards to the highest acceptable probability that an invalid message is recognized as valid and the number of maximum invalid messages before the key used to authenticate the message is retired is given in Annex A of [14]:

$$Tlen \geq log_2 \left(\frac{MaxInvalids}{Risk} \right)$$

For example, a MAC of 1 byte is sufficient to guarantee a probability of 2^{-7} if the key is retired after 2 invalid messages are detected. An analysis of the bits required to tolerate a certain number of invalid messages and the number of old packets to keep in memory is given in [15]; here the authors extend the idea to different MAC in a same message, making it suitable for multicast authentication. The limitation of message size could be addressed using a transport protocol, but this will result in an increase of the busload, in addition to a greater latency given by the reassembly of the data.

In existing standards like SAE J1939 [9] or ISOBUS [10], the majority of messages do not have room for a MAC, since all the 64 bits of CAN are used. In this kind of messages, proper authentication is considered not possible using the standard CAN.

Authentication schemes such as Leia [16] provide lightwight authentication and require the use of an additional CAN message for each message to be authenticated. While this could be acceptable for highly critical messages, its use of the CAN ID is not compatible with ISOBUS. The importance of using the CAN ID here lies in the transmission of a counter related to the authentication protocol and to operation like synchronization between ECUs.

Other authentication mechanisms have been proposed for extensions of the classic CAN bus like Flexray and CAN-FD, see for example [17] for a comparison.

B. Integrity over CAN bus

Message integrity is normally achieved on a CAN network with a CRC16 checksum. While CRC16 is not suited for cryptographic purposes, since for example it is a linear function and it is easily invertible, it is not easy to modify a message in transit without being detected, because of the bus nature of the CAN network. A replay attack, however, is feasible if messages are not authenticated, since an attacker can inject packets in the network pretending to be anyone, also with higher priority that legitimate packets. Additional checksum or hash algorithms can be used; however in some standard messages (e.g. SAEJ1939 TSC1 - Torque Speed Control 1) there is no room for additional integrity checks. The possibility to increase the integrity protection of the message depends then on the specific data and the admissible data ranges, which can be given by the protocol (e.g. command codes) or physical meaning (coherency control).

C. Confidentiality over CAN bus

The most common way to achieve confidentiality on a communication is using some form of encryption. Given the security requirements listed in section III, a natural choice for CAN bus is a symmetric block cypher, which block length is exactly 64 bits; algorithms like DES, Blowfish are then well suited for this purpose. A schema of the encryption and decryption procedure is visible in Figure 1. Generally, for short term security a symmetric key of 96 bit is sufficient, raising to 112 bit for middle-term security and 128 bit for long-term security, as reported in Table I. However, for real-time traffic, a 64 bit symmetric key size might be enough, if we consider a key refresh procedure. Another interesting class of algorithms to consider is the class of stream ciphers, which allow the encryption of exactly the number of bytes needed. However, this needs further considerations, since this class of algorithms is often weak against sniffing attacks. This can be mitigated adding a time-variant component to a message, for example xor-ing the plaintext with a time-varying nonce. A greater security level, intended as an increased key size, can be obtained using a symmetric block cipher in Counter mode (CTR) mode, as shown in Figure 2. Here the plaintext data are not directly encrypted, but they are xor-ed with the result of encrypting a known value, called nonce. However this come to a price, that is the involved ECUs must have a shared knowledge of a time-variant value, such as a running number, which must be used to construct a nonce that must be different for each encryption in order to guarantee the security of CTR mode. A distributed running counter such as the one used in [18] can be used. The key exchange procedure will need a transport protocol in this case; in order to achieve a sufficiently fast key exchange and computation, it is essential to use short key sizes. For this reason, the best algorithm to use for key exchange is an Elliptic Curve algorithm with a key size that will be a compromise between the security grade and the overhead (bus and computational) required.

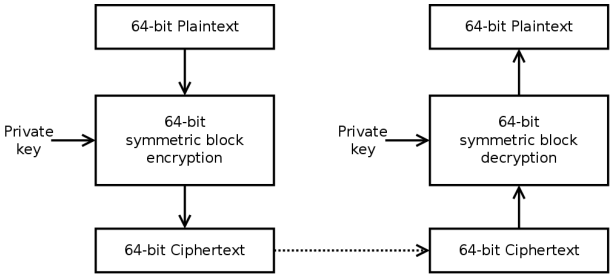
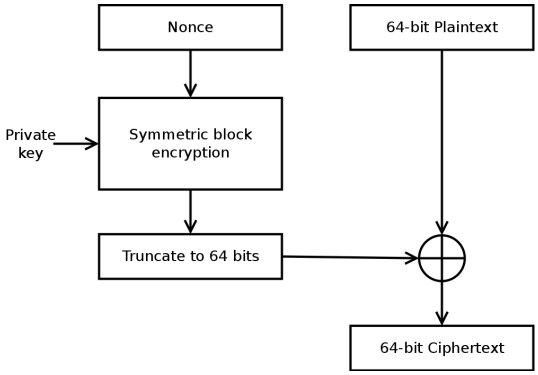
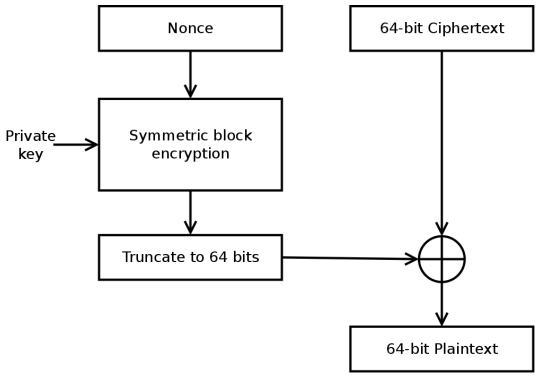


Fig. 1. Encryption and decryption procedure.



(a) Encryption with CTR mode.



(b) Decryption with CTR mode.

Fig. 2. Encryption (a) and decryption (b) procedure with CTR mode.

V. Future - High-speed network (Ethernet)

Modern Ethernet network are designed to permit a greater packet size, up to *1500 bytes*, if there are no additional tags such as 802.1q or 802.1ac. Furthermore the topology can be both a physical bus (using hubs to interconnect the ECUs) and a segmented network (using switches to separate the collision domains); the possibility of using a switch or router is essential to mitigate *Replay*, *Sniffing DoS* and *Black Hole* attacks, since they can filter the traffic only to the links directed to the source and destination ECUs; furthermore, it is possible to create different VLANs according to IEEE 802.1q [20] to logically separate different network segments, which can physically share the same link.

Security Level	Symmetric	ECC	RSA
Very short term, weak attacker model	64	128	816
Short-term (Standard level)	96	192	1776
Middle-term	112	224	2432
Long-term	128	256	3248

TABLE I
Recommended key length in bits for asymmetric-key and symmetric-key cryptography [19]. Additional references are available at http://www.keylength.com.

A. Authentication over Ethernet

In Ethernet-based networks, authentication can be performed at different levels. Here the packet length can be much greater than CAN networks, so for single messages a MAC code can be used.

The use of an asymmetric key to authenticate the messages, for example with a digital signature scheme, is not a viable solution, given the requirements listed above. Although asymmetric cryptography, in particular ECC, has been shown to be feasible on constrained embedded platforms, for example in [21], the encryption and decryption time still is not sufficiently low for real-time traffic, where the maximum time allowed for generating an authentication token can be the order of milliseconds. For this reason, the best approach is to use a Message Authentication Code (MAC), which requires a key exchange procedure. One example of such algorithms is HMAC [22]. Asymmetric key cryptography is however still valid if used as part of the key exchange and key refresh procedures, since in this case it is admissible to have a low-priority task which takes hundreds of milliseconds to terminate.

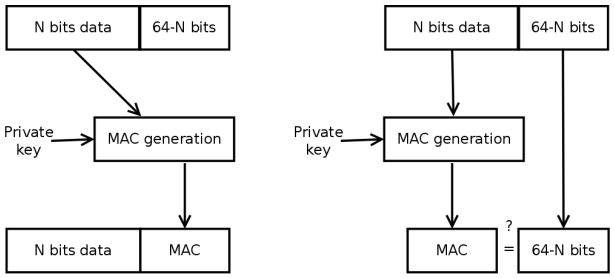


Fig. 3. Authentication and Verification procedure with MAC code procedure.

B. Integrity over Ethernet

On an Ethernet network, data integrity is only assured on a link basis with a 32-bit CRC; this is due to segmentation. While one could argue that a switch could not modify the CRC, in practice there is no guarantee, and a switch could also regenerate the CRC for all egress packets; in this case, an error during the transit of the packet from the ingress queue to the egress queue would cause an integrity loss. In other words, this is only a hop-to-hop integrity check. End-to-end integrity is usually provided by higher layer protocols, for example by UDP and TCP [23]. In general, Ethernet allows for higher integrity levels to be reached, since it poses no limitation on

the integrity check to use, and there are many known protocols used on IP-based networks, which can be used for this purpose. However, since most protocols use CRC, it is not sufficient to protect from deliberate modification of a message; in this case, authentication is necessary.

C. Confidentiality over Ethernet

The maximum payload contained in a single Ethernet frame is of 1500 bytes, so an algorithm like AES can be used, with the padding considerations listed above. Moreover, the key exchange procedure does not need a transport protocol since, if sufficiently short key sizes are used, for example using an Elliptic Curve algorithm, the messages to be exchanged are easily contained in a single packet.

VI. Discussion

As analysed so far, the opportunities that attackers have in the widely used CAN-networks are not negligible. The fault does not totally belongs to the CAN design, but its weaknesses come from the fact that the CAN system was not thought to be applied in a communication domain such as the Internet. Thus, an attacker, who acts as depicted in §II, can achieve his/her goals with no particular effort since mostly of the basic security properties miss. In fact, on a CAN network, due to the limited message length, a sufficiently secure authentication method is not feasible; if a full 64-bit standard message is required, an authentication method is not possible at all. On the other hand, on Ethernet it is possible to securely authenticate a message, at the cost of a little overhead given by the addition of the MAC. Again, the security of the MAC is limited by the computational power available on an ECU, but since most microcontrollers embed a hardware implementation of a symmetric block cypher, which can be used in CBC-MAC or CMAC mode, a secure authentication is immediately possible. If no hardware cryptographic module is present, a software implementation of such symmetric block cypher can be used; some comparison between the time required for different algorithms is presented in [24], [25] and [26].

A. Open Issues

There are many additional issues that need to be addressed in order to build a secure vehicular network. The validity of CAN messages, for example the TSC1 requesting a certain value of torque or RPM, could be verified with appropriate coherency checks, based on past and/or reasonable future values. How to consider past values and calculate reasonable values is however an issue that shall be treated on a field-by-field basis, and is still an open issue. An essential step for using cryptographic algorithms is having a pre-shared key (for symmetric key algorithms) and a certificate management policy (for asymmetric key algorithms). Possible solutions include prior key distribution, having a master key used to derive a session key, having a dip switch, which can be used to configure a cryptographic key for an ECU, or using a Trusted Platform Module (TPM) for key management. Finally, the initialization of the vehicular network, if required, must consider and eventually include key management and setup. Depending on the policy, the network can tolerate an initial unsecured initialization or rely on a Public Key Infrastructure (PKI) for key exchange. This problem is currently being considered in the TIM working group (Tractor Implement Management) of AEF (Agricultural Industry Electronics Foundation).

VII. Conclusions

The increasing number of external connections make cars and heavy duty vehicles vulnerable to cyber-security threats. The current communication system, like CAN-bus, was not designed to be ready to security threats that may impact to the safety of drivers and passengers. In this paper, we have analysed the CAN network bus, and we have studied it against some well-known security attacks. Seen the lack of security of the CAN-bus, we have proposed some solutions that may be applied to CAN-bus to make it more robust against the security threats. However, not all solutions can be implemented seen the nature of the CAN-bus, but also due to the limited hardware resources of the ECUs. Then, we have analyzed the Ethernet protocol against the same security threats, and we have showed how the Ethernet protocol may guarantee a higher level of robustness compared to CAN.

References

[1] C. Valasek and C. Miller, "Adventures in automotive networks and control units," in *DEFCON 23*, 2015.

[2] ——, "After jeep hack, chrysler recalls 1.4m vehicles for bug fix," Online, 7 2015. [Online]. Available: https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix/

[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 *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, ser. SP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 447–462. [Online]. Available: http://dx.doi.org/10.1109/SP.2010.34

[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 Conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=2028067.2028073

[5] C. Valasek and C. Miller, "A survey of remote automotive attack surfaces," IOActive Technical White Paper, Tech. Rep., 2014.

[6] ——, "Hackers remotely kill a jeep on the highway. with me in it," Online, 7 2015. [Online]. Available: http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[7] M. Al-Shurman, S.-M. Yoo, and S. Park, "Black hole attack in mobile ad hoc networks," in *Proceedings of the 42Nd Annual Southeast Regional Conference*, ser. ACM-SE 42. New York, NY, USA: ACM, 2004, pp. 96–97. [Online]. Available: http://doi.acm.org/10.1145/986537.986560

[8] ISO, "Road vehicles diagnostic communication over controller area network (docan)," The International Organization for Standardization, Genève, Switzerland, Tech. Rep. ISO 15765, 2011.

[9] SAE, "Recommended practice for a serial control and communications vehicle network," Society of Automotive Engineers, Warrendale, PA, Tech. Rep. SAE J1939, 2013.

[10] ISO, "Tractors and machinery for agriculture and forestry – serial control and communications data network - part 3: Data link layer," The International Organization for Standardization, Genève, Switzerland, Tech. Rep. ISO 11783, 2014.

[11] H. Kleinknecht, "Can calibration protocol version 2.1, technical report," Standardization of Application/Calibration Systems task force, Tech. Rep., 1999.

[12] ISO, "Road vehicles - diagnostic systems - keyword protocol 2000 - part 3: Application layer," The International Organization for Standardization, Genève, Switzerland, Tech. Rep. ISO 14230, 2013.

[13] A. V. Herrewege, D. Singelee, and I. Verbauwhede, "Canauth - a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT Workshop on Lightweight Cryptography*, November 2011, pp. 229–235.

[14] M. J. Dworkin, "Sp 800-38b. recommendation for block cipher modes of operation: The cmac mode for authentication," National Institute of Standards and Technology, Gaithersburg, MD, United States, Tech. Rep., 2005.

[15] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 165–174.

[16] A.-I. Radu and F. D. Garcia, "Leia: A lightweight authentication protocol for can." in *21st European Symposium on Research in Computer Security (ESORICS 2016)*, 2016.

[17] 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, pp. 7:1–7:8. [Online]. Available: http://doi.acm.org/10.1145/2818362.2818369

[18] 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 Systems Letters*, vol. 7, no. 1, pp. 11–14, March 2015.

[19] ECRYPT, "Yearly report on algorithms and keysizes," ECRYPT II, Tech. Rep., 2012.

[20] "IEEE standard for local and metropolitan area networks–bridges and bridged networks," *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)*, pp. 1–1832, Dec 2014.

[21] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe, "High-speed curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers," *Des. Codes Cryptography*, vol. 77, no. 2-3, pp. 493–514, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1007/s10623-015-0087-1

[22] H. Krawczyk, M. Bellare, and R. Canetti, "Hmac: Keyed-hashing for message authentication," Internet Requests for Comments, RFC Editor, RFC 2104, February 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2104.txt

[23] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Trans. Comput. Syst.*, vol. 2, no. 4, pp. 277–288, Nov. 1984. [Online]. Available: http://doi.acm.org/10.1145/357401.357402

[24] Y. W. Law, J. Doumen, and P. Hartel, "Survey and benchmark of block ciphers for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 65–93, Feb. 2006. [Online]. Available: http://doi.acm.org/10.1145/1138127.1138130

[25] C. Mickael and M. Kevin, M.and Marine, "Survey and benchmark of lightweight block ciphers for wireless sensor networks," *IACR Cryptology aPrint Archive*, 2013.

[26] D. S. A. Elminaam, H. M. A. Kader, and M. M. HadHoud, "Performance evaluation of symmetric encryption algorithms," *Communications of the IBIMA*, 2009.