LaaCan: A Lightweight Authentication Architecture for Vehicle Controller Area Network

Syed Akib Anwar Hridoy(✉) and Mohammad Zulkernine

Queen's Reliable Software Technology (QRST) Lab, School of Computing, Queen's University, Kingston, ON, Canada
{akib.anwar,mz}@queensu.ca

Abstract. Vehicle manufacturers are installing a large number of Electronic Control Units (ECU) inside vehicles. ECUs communicate among themselves via a Controller Area Network (CAN) to ensure better user experience and safety. CAN is considered as a de facto standard for efficient communication of an embedded control system network. However, it does not have sufficient built-in security features. The major challenges of securing CAN are that the hardware of the ECUs have limited computational power and the size of a CAN message is small. In this paper, a lightweight security solution, LaaCan is designed to secure CAN communication by adopting the Authenticated Encryption with Associated Data (AEAD) approach. The architecture ensures confidentiality, integrity, and authenticity of data transmission. The experimental results show that the delay of LaaCan can be reduced depending on hardware configurations. We consider it lightweight since it adds a low overhead regardless of performing encryption and authentication. We evaluate LaaCan using four metrics: communication overhead, network traffic load, cost of deployment, and compatibility with CAN specification. The evaluation results show that the proposed architecture keeps the network traffic unchanged, has low deployment cost, and is highly compatible with the specification of the protocol.

Keywords: CAN bus · In-vehicle network security · AEAD

1 Introduction

Vehicles were considered as mechanical machines before the introduction of software inside them. Components such as engines, brakes, and gears were assembled into a car in coherence with the principle of mechanics. Yet, the limited accuracy of mechanics led to undetectable failures, and vehicle safety was in threat. The automotive industry moved towards the adaption of digital electronics and software controls in the vehicle to improve the scenario. Manufacturers started installing electronic sensors in vehicles for driving safety and assistance. The

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
N. Park et al. (Eds.): SecureComm 2020, LNICST 336, pp. 215–234, 2020.
https://doi.org/10.1007/978-3-030-63095-9_12

automotive industry introduced Electronic Control Unit (ECU) in 1970 to collect information from the sensors and control the mechanical components. An ECU can request another ECU for its sensor information to make a collective decision. These ECUs form an in-vehicle network to communicate with each other. For in-vehicle communications, the most widely used medium is the Controller Area Network (CAN).

With the revolution of ECUs, new features are added to vehicles to enable them to make intelligent decisions. These features provide autonomous driving support as well as safety and convenience to users. However, they expose the previously isolated vehicle system to cyberspace, which introduces the opportunity of cyberattacks. These attacks endanger the privacy and safety of a vehicle.

Attacks try to control vehicle functionalities illegally. ECUs are responsible to control these functionalities and they communicate via a CAN bus. Therefore, these attacks highly relate to CAN communications and the security of these communications must be a concern. The two most significant purposes of CAN development were to reduce the wiring complexity and cost. At that time, the security of communication between vehicular components was not a concern as a vehicle was a closed system without communications with other devices or vehicles. Hence, the automotive engineers implemented CAN following the concept of broadcast-based serial communication. As a result, any ECU connected to the network can read or send messages.

In-Vehicle Infotainment (IVI) system is connected to the CAN bus. It increases the security risks as IVI connects external devices through the wireless medium such as Bluetooth and Wi-Fi. There are third-party applications available for IVI to provide entertainment and navigation services [18]. Besides, third party dongles can be plugged into the OBD-II diagnostic port to monitor the status of vehicle systems such as the engine and transmission. These dongles connect to smartphones via Bluetooth. A malicious application installed on a phone that is connected to the OBD-II dongle can help the attacker to read the network traffic [33]. The reverse engineering of recorded communication may lead to an attack. The lack of confidentiality, integrity, and authenticity features in the CAN protocol are the reasons for these attacks.

In this paper, we design LaaCan, a security architecture that implements a lightweight authenticated encryption based on a pre-shared secret key to assure confidentiality, integrity, and authenticity. Authenticated encryption ensures the privacy and authentication of data. It is implemented by adopting Authenticated Encryption with Additional Data (AEAD) cipher. To find the best applicable AEAD cipher in CAN, we explore five AEAD-based ciphers and the analysis shows that ChaCha20-Poly1305 has the best credibility in low powered ECUs. ChaCha20-Poly1305 authenticates data using Message Authentication Code (MAC) that has to be transmitted with the message. The CRC field in CAN frame is used for error detection and error detection is part of data integrity process [23]. Since LaaCan ensures data integrity, we replace the CRC data with MAC that helps to maintain unchanged network traffic and low overhead. The experimental results show that LaaCan provides strong security with low com-

Table 1. Standard CAN frame description.

Field	Length	Description
Start of frame	1 bit	Indicates the beginning of a frame
Arbitration	12 bits	Contains the type and priority of the message
Control	6 bits	Includes the length of the data
Data	64 bits	Holds the transmitted data
CRC	16 bits	Cyclic redundancy check field used for detecting error of the transmitted data
ACK	2 bits	Acknowledges reception of valid CAN messages
EOF	7 bits	Specify the end of the frame

munication overhead and protects the network from the most common form of attacks. The evaluation results show that LaaCan does not have any effect on network traffic load, has low deployment cost, and is highly compatible.

The remainder of the paper is organized as follows. In Sect. 2, we provide an overview of CAN protocol and discuss the related work by providing a classification of existing solutions. Section 3 presents the design of LaaCan. The implementation details and evaluation results are described in Sect. 4. We conclude the paper in Sect. 5 with a discussion on limitations and future work.

2 Related Work

In this section, we discuss the existing CAN security solutions. Here, we compare and contrast LaaCan with the related work qualitatively. Before presenting the related work, we briefly discuss the Controller Area Network (CAN). CAN is a multi-master broadcast-based bus system with a bandwidth up to 1 Mbit/s. It is widely used for embedded system communication as it is efficient and cost-effective. In standard CAN protocol, broadcast messages do not contain any receiver information [7]. Therefore, adding and removing nodes is easier. Table 1 describes each field of a CAN message.

Considering the different implementation techniques of security countermeasures, we present the related work as a classification of existing security solutions. We classify the security measures in terms of security enforcement procedures. Figure 1 depicts the classification. The communication is primarily secured by adopting cryptographic architecture and intrusion detection systems (IDS) or both. The cryptographic architectures mostly involve different ways of encryption and authentication mechanisms to secure network transmission from adversaries. A number of works in this category are discussed in the following subsections. In contrast, the IDS learn the predefined activities and policies to detect

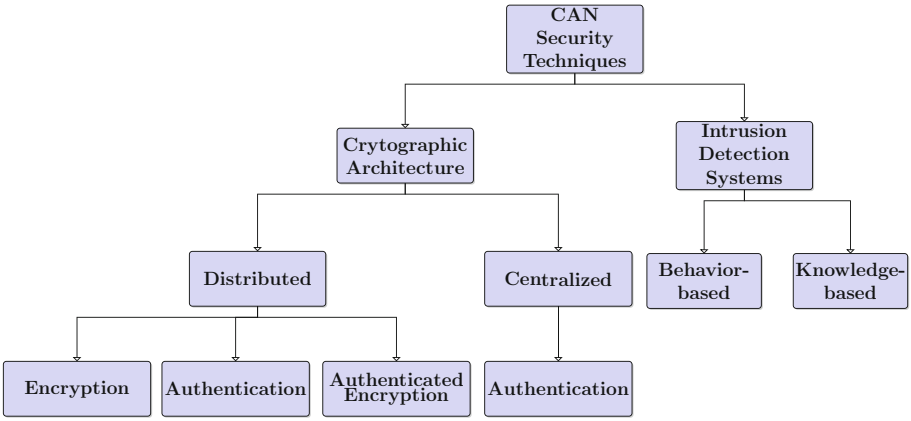

Fig. 1. Classification of CAN security solutions.

and report any malicious activity. The existing IDS of in-vehicle networks can be categorized into behavior-based and knowledge-based techniques [16]. For the sake of space limitation and relevance, we do not discuss the IDS further.

2.1 Distributed Approach

In the distributed cryptographic architecture, security features are implemented inside ECUs and there is no need for a central authority. Although an ECU gets compromised by adversaries, they cannot control the full network. The distributed methods can be further divided into three categories: encryption, authentication, and encrypted-authentication. The encryption-based approaches involve data encryption for security purposes; however, it does not include an authentication mechanism. On the other hand, authentication-based approaches only authenticate data. These approaches lack confidentiality as they do not encrypt data. The authenticated encryption approaches encrypt and authenticate data. Though it may require more processing time, authenticated encryption provides strong security. We present the related work for each category in the following paragraphs.

Distributed Encryption. CANTrack [14] only encrypts the message using the message counter to prevent the replay attacks. It does not include any authentication mechanism. All ECUs maintain a message counter for each message type and use the counters as encryption keys.

Distributed Authentication. VeCure [32] introduces trust-based grouping of ECUs along with Message Authentication Code (MAC) for authentication purposes. It divides ECUs into two groups. ECUs with external interfaces such as OBD-II and infotainment system are assigned to low-trust group. The other

ECUs with no external interfaces are assigned to high-trust group. Authentication is only performed for the communication of high-trust ECUs. They share a common secret key to generate MAC. MAC is transmitted using an *extra message*, which increases network traffic and message processing time. LaaCan does not affect the network traffic as no additional message is used for MAC transmission. Herrewege et al. [31] uses HMAC (Hash-based Message Authentication Code) to generate the message authentication code using shared secret keys. CAN+ [35] is used to share secret keys as it allows additional 120 bits to be attached to a frame. Each message type has a unique secret key, and all secret keys are stored in the participating ECUs. Therefore, compromising of an ECU can reveal all secret keys. Though we use a shared secret key, LaaCan ensures message freshness by involving a counter. The message freshness value makes it harder for attackers when the secret key is compromised as attackers need to keep track of freshness value. LiBrA-CAN [15] authenticates messages using Mixed-Message Authentication Codes (M-MACs). It divides the ECUs into multiple small groups. Each group of ECUs share a secret key. The authentication is performed by employing a helper ECU to compute MAC. Both the sender and the helper computes MAC partially. The receiver ECU performs authentication by merging the partially generated MAC. LiBra-CAN has high communication overhead and uses CAN+ [35], which is not well recognized. Therefore, it is not compatible with the standard CAN.

Distributed Authenticated Encryption. Woo et al. [33] assure data confidentiality, integrity, and authenticity based on MAC. The MAC is transmitted using both the extended ID field and the CRC field. Thus, the payload of the network stays unchanged after the initial key distribution. However, a widely used standard SAE J1939 [17] uses the extended ID field to transmit Parameter Group Number (PGN). Hence, reserving extended id field for MAC excludes existing standards. LASAN [22] ensures three security features as well. A centralized security module authorizes all the general ECUs. AES-128 is used in Cipher Block Chaining (CBC) mode. AES requires hardware support and resource constraint ECUs do not have hardware acceleration support for cryptography [11]. Also, the installation of a central ECU brings hardware change, which increases the deployment cost. LaaCan does not require any hardware support and is adaptable by software updates. TOUCAN [8] assures confidentiality, integrity, and authenticity. It also applies AES-128 for data encryption, and Chaskey [20] for MAC computation. The data payload is reduced to 40 bits to fit 24-bit MAC. However, reducing the size of payload makes it incompatible to existing designed protocols that make use of the full length of the payload. Alam et al. [5] introduce identity-based access control for ECU authorization. The symmetric key cryptography and digital signature are applied to assure confidentiality, integrity, and authenticity. The symmetric keys are shared using elliptic curve-based Public Key Encryption (PKE). Though digital signature ensures non-repudiation, it is slow. LaaCan falls in the category of distributed authenticated encryption. The AES cryptographic algorithm is used by [33] and [8]. AES consumes more resources and requires hardware support to reduce the processing delay. Thus,

AES is not suitable for resource-constrained ECUs. Besides, we avoid using the extended id field to transmit authentication code to assure backward compatibility. We also refrain from reducing payload size, which makes LaaCan adaptable to existing implementations.

2.2 Centralized Approach

In a centralized cryptographic architecture, data is authenticated by central ECU. There is no involvement of encryption in a centralized approach. Although these approaches may involve some processing in participant ECUs related to authentication code generation, the authentication process is served by the central ECU. CaCAN [25] installs a monitor ECU in the bus. The monitor ECU compares both the received and calculated MAC for message authentication. TCAN [9] authenticates based on the physical location of the ECU in the network. Two dedicated node repeater and monitor are used to identify the physical location of ECUs from the message reception time difference. Although TCAN authenticates the messages using the physical location, it involves cryptography for the communication between the general ECUs and the monitor in the initialization phase. CaCAN [25] and TCAN [9] lack confidentiality as messages are not encrypted. Moreover, installing dedicated ECUs for authentication is expensive and not adaptable to already manufactured vehicles. Also, they are vulnerable to man-in-the-middle (MITM) attacks.

3 LaaCan Design

In the preceding section, we discussed existing CAN security solutions and their shortcomings. Among the different approaches of cryptographic implementations, only the distributed authenticated encryption architecture is capable of ensuring confidentiality, integrity, and authenticity. Intrusion detection systems highly depend on the trained model, which is heavy computation for resource constraint devices. Considering the limited resources of ECUs and the consequences of in-vehicle security attacks, we design a lightweight security solution by adopting distributed authenticated encryption architecture.

In this section, the architecture of LaaCan is presented. Since a security architecture should be integrable with the existing standards, the design is analyzed to verify the compliance with standard CAN and AUTOSAR [1]. Later, the design challenges and solution options are explained.

3.1 Authenticated Encryption Design

We propose a lightweight authentication architecture for CAN communication. We implement Authenticated Encryption with Associated Data (AEAD) scheme as it is capable of ensuring confidentiality, integrity, and authenticity. We identify available AEAD schemes used in industry and compare the performance to determine the best fit in CAN protocol. The experimental analysis shows that

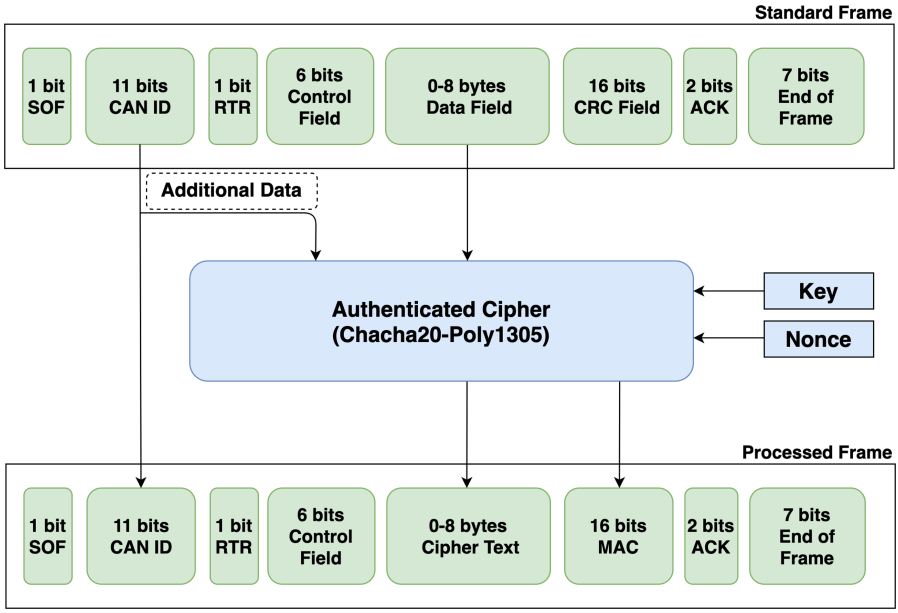

Fig. 2. Design of the authenticated encryption.

among five widely used AEAD algorithms, ChaCha20-Poly1305 has the best performance based on the communication overhead and security measures.

Figure 2 shows the architecture of LaaCan. AEAD requires four inputs: a secret key, nonce, plain text, and additional data. The secret key has to be random. The nonce is used to keep the keystream unique for each operation. Both the sender and the receiver have to know the secret key and nonce. The outputs of AEAD are ciphertext and MAC. The ciphertext is privacy protected plain text, and MAC is a tag to ensure data accuracy.

In the first phase, the message is encrypted using the ChaCha20 stream cipher to assure confidentiality. For encryption, ChaCha20 generates a block using the key block, nonce, and constant characters. Then, it splits the block to form a matrix and runs 20 rounds of alteration between cells. Each round performs four quarter-rounds to process both column and diagonal values. Every quarter-round involves bit operations of add, XOR, and rotate. After the 20 rounds, the matrix is serialized to generate a keystream block. The ChaCha20 performs XOR operation with plain text to generate ciphertext [24]. The keystream block generation process does not involve the plain text. Therefore, it is possible to generate a keystream block earlier to reduce the encryption process delay. After sending a message, a keystream is generated for the next message, which is referred to as a forward key generation. Thus, during the encryption of the next message, the keystream block will be ready to XOR with plain text. Since the forward keystream generation is performed, a keystream block is generated

during initialization for the first message. Forward keystream generation reduces the delay by 15% 16 MHz micro-controllers for each message transmission.

The message authentication is necessary to ensure authenticity and integrity. The authentication scheme generates MAC to achieve message authentication. However, each CAN message has an identifier, and alteration of it is considered as an attack. As a result, the authentication of the message identifier is necessary. AEAD provides authentication for associated data. Hence, we feed the message identifier as additional data to AEAD. We use the message counter as the nonce, which gives freshness to the secret key. The ECUs have non-volatile storage to store system and user data [34]. The freshness value can be stored in the non-volatile storage so that it can be retained in each run. Since CAN is a broadcast-based protocol, all the ECUs receive all transmitted data in the network. Therefore, it is not required to attach the counter with a transmitted message. Besides, CAN protocol has a message filtering feature, which uses the identifiers for filtering. Therefore, we do not encrypt the message identifier. However, if the identifier is altered during the transmission, the authentication fails.

Due to the limited payload, CAN protocol cannot accommodate additional data bits in the message. Message encryption using ChaCha20 does not increase the length of the encrypted message. However, MAC has to be transmitted to the receiver to complete the authentication process. Wang et al. [32] used an extra message to carry MAC. Additional message transmission increases the traffic of the network. It also increases the delay in message processing as the receiver has to wait for the MAC. Moreover, no other ECU can communicate between the transmission of the original message and MAC. CAN protocol uses a 15-bit CRC field for error detection of transmitted data. Since LaaCan ensures the integrity of data and error detection is part of data integrity [23], we use the CRC field to attach a 15-bit authentication code. Two existing work [10,33] demonstrated a similar approach of using the CRC field to transmit MAC and showed the feasibility of the solution.

We follow the distributed approach of ensuring security where the network traffic is encrypted and authenticated by all the ECUs. MAC is generated and appended to the message by a sender ECU. There is no need for a central ECU, which allows integrating the solution to existing vehicles with software updates only. Furthermore, LaaCan does not transmit any extra message for security purposes. It does not involve any modified version of the standard protocol also. Therefore, it is feasible to integrate it with the standard CAN protocol as well as standards widely implemented, such as SAE J1939 [17].

3.2 Design Requirement Analysis

To make LaaCan adaptable with the existing systems and standards, we identify two major design requirements: standard CAN and AUTOSAR compliant. The analysis of the design requirements is discussed in the next paragraphs.

Table 2. AUTOSAR profile of LaaCan.

Parameter	Configuration value
Algorithm	ChaCha20-Poly1305
Length of Freshness Value (parameter SecOCFreshnessValueLength)	0 bit
Length of Truncated Freshness Value (parameter SecOCFreshnessValueTxLength)	0 bit
Length of Truncated MAC (parameter SecOCAuthInfoTxLength)	15 bits

Standard CAN Compatibility
The CAN protocol is standardized under ISO (International Organization for Standardization) 11898-1 [2] that describes the data transmission process and message format. The compliance with standard protocol gives solution flexibility of integration with the majority of the existing systems and standards. A security solution should be able to run on existing ECUs and networks to achieve the standard CAN compliance. Some security solutions [15,31] use CAN+ [35] protocol that requires hardware change in transceiver [8]. LaaCan does not require any hardware changes. Unlike some other related work [9,15,22,25,30], we do not install any dedicated ECU. Therefore, we claim LaaCan is standard CAN compliant.

AUTOSAR Compatibility
The AUTOmotive Open System ARchitecture (AUTOSAR) [1] is a development partnership of automotive stakeholders to implement standardized software architectures for ECUs [3]. The specification of AUTOSAR describes standard development practices. To verify the compliance with the AUTOSAR standard, we study the release document of AUTOSAR 4.3.1, Specification of Secure Onboard Communication [6]. According to the specification, sender and receiving nodes require to maintain freshness value. Though, it is not mandatory to add the freshness value to the payload, it has to be considered during the MAC generation. LaaCan maintains a message counter to assure message freshness. In the standard CAN protocol, all the broadcasted messages are received by the participating nodes. Thus, there is no need to add the freshness value to the payload as all the ECUs can maintain the counter. AUTOSAR applies security solution based on the security profile. Table 2 shows the security profile of LaaCan for AUTOSAR. The first parameter "Algorithm" stands for the name of the cryptography algorithm, which is ChaCha20-Poly1305, in this case. SecOCFreshnessValueLength and SecOCFreshnessValueTxLength parameters are related to the freshness value. The configuration values for these parameters are 0 bit as the freshness value is not added to the payload. SecOCAuthInfoTxLength denotes the length of the truncated MAC, and we use 15-bit of the truncated MAC.

3.3 Design Challenges and Solutions

There exist three main challenges in the design of LaaCan: choosing the cryptographic algorithm, deriving the shared secret key, and fixing the MAC size and transmission process. We discuss these challenges and solution choices in the next paragraphs.

Choosing Cryptographic Algorithm
We use authenticated ciphers to ensure data confidentiality, integrity, and authenticity. There are several authenticated ciphers available. However, we need to select one that provides strong security with efficient performance in ECUs.

The in-vehicle communication has to be real-time. However, security measures add a delay in communication. The delay highly depends on the hardware components on which the security solution is running. The vehicle ECUs are resource-constrained. Thus, a solution must have strong security with low communication delay, less memory consumption, and less power consumption. We choose widely used authenticated ciphers and compare the execution time to encrypt and authenticate 8 bytes of data. We take 8 bytes of data because a single CAN frame can have up to 8 bytes of data. We shortlist AES-GCM, Speck-GCM, ChaCha20-Poly1305, Ascon, and Acorn for the comparison as they are widely used. We run the them in three different hardware configurations. We discuss experimental setups in detail in Sect. 4.1.

Figure 3 shows the execution times of the shortlisted ciphers. Chacha20-Poly1305 has the lowest execution time in all three configurations. It is significantly faster than the most widely used AES-GCM. The AES-based ciphers perform better when the hardware has AES support. However, for resource-constrained devices without hardware support, Chacha20-Poly1305 takes less time while consuming low memory and power than some widely used authenticated ciphers [13].

Shared Secret Key
The cryptographic algorithms use a secret key for data encryption. The key selection is essential as the security strength depends on it. The same key can be used for encryption and decryption in case of symmetric-key cryptography. In asymmetric key cryptography, one key is used in the encryption process and a different key is required for the decryption process.

The asymmetric key technique is significantly slower than symmetric key and consumes more resources [27]. Hence, we choose the symmetric key cryptography. In symmetric-key cryptography, the sender and receiver need to share a common secret key. However, key generation and sharing between ECUs is complex as the network is multicast-based, and there is no central authority. If the same key is used throughout the lifetime of the vehicle, an adversary can analyze the network traffic to retrieve the secret key. Therefore, we generate a session key at the starting of a vehicle. To generate a session key, we use a long term symmetric key and a session id of 128 bits each.

A 128-bit session-id allows 2^{20} or 1.04 million distinctive sessions, which are enough for a vehicle lifetime. At the starting of a vehicle, the session-id incre-

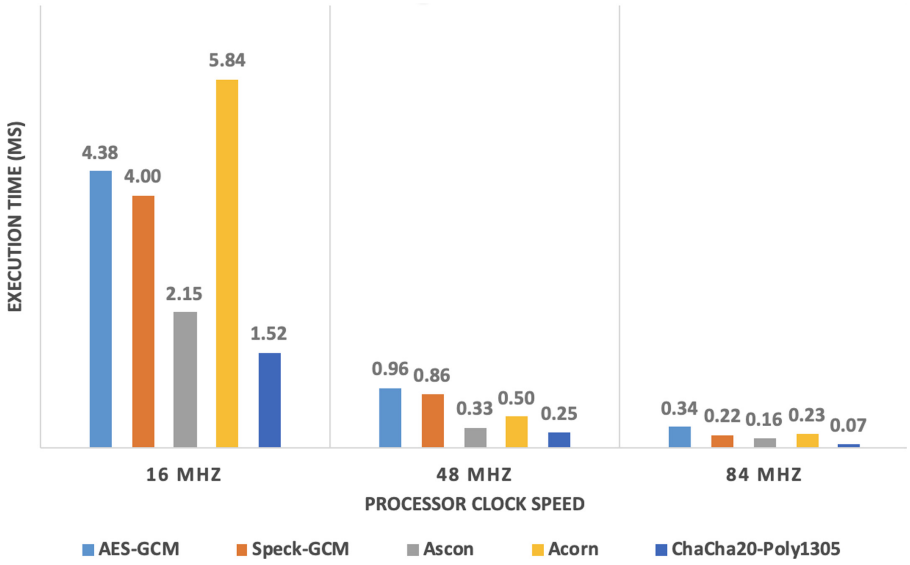

Fig. 3. Execution times of authenticated ciphers.

ments and a session key is generated. ChaCha20 generates the session key using the session id and long term symmetric key. A new session key is also generated when the message counter value reaches its limit. The key establishment and share process are resource consuming and increases the network traffic, which we avoid by deriving the session key. We use the session key for a limited time. Hence, even if an adversary retrieves a key by analyzing network traffic, it can bypass the security measures for only one session as session key changes every time the vehicle starts. The session key generation and key setup process require a maximum of 1.83 ms, which is very low. As it is only done at the starting of the vehicle, it is acceptable.

MAC Size and Transmission

LaaCan authenticates messages based on MAC. The security strength of MAC depends on its length. A MAC with a bigger length has better security strength. However, the size of the CAN frame is limited. CAN frame has a 15-bit CRC field, which provides error detection support only. CRC check can recognize an error that occurs during transmission. LaaCan ensures data accuracy based on MAC. The MAC authentication at the receiver end fails if any bit in the original data changes due to a noisy transmission channel. Therefore, we decide to replace CRC with MAC. The replacement of CRC with MAC allows keeping the traffic of the network the same as standard CAN. Otherwise, data payload has to be reduced [8], or additional message transmission is required [32]. Woo et al. [33] and Bittl [10] also propose overwriting CRC field with MAC.

The probability of guessing MAC is 2^{-mac_length} [20]. We generate 128-bit MAC and truncate it to 15 bits for transmission. Therefore, the probability of

guessing MAC is 2^{-15}. In other words, an attacker requires 2^{15} or 32,768 attempts to guess the MAC that is not feasible considering the limited resources available in ECUs.

4 Implementation and Experimental Evaluation

To assess the effectiveness of LaaCan, we validate the security features through experiments and preform security analysis against attacks. In this section, the implementation details are provided and the security analysis is discussed. Later, the evaluation results and comparison with some of the existing work are presented.

4.1 Implementation

LaaCan has to be integrated into Electronic Control Unit (ECU) software by installing in the ECU micro-controller. A micro-controller works as a core processing unit of an ECU. Several exiting work [8,25,30,33] used micro-controllers for experiments. To simulate an in-vehicle network consisting of multiple ECUs connected by a CAN bus, we connect multiple Arduinos by the CAN bus. An Arduino is a programmable micro-controller. However, it does not have the capability of CAN transmission. Hence, we install CAN-Bus Shield V2.0 [28] on top of Arduino that provides CAN transmission ability to Arduino.

Micro-controllers come in different hardware configurations. CAN security researchers mostly use ARM-based micro-controllers with CPU clock speeds between 40–150 MHz for their experiments. However, AVR-based microcontrollers are widely used in vehicle ECUs. Therefore, we consider microcontrollers with three different configurations, which are Arduino Uno (16 MHz 8-bit AVR), Arduino Zero (48 MHz 32-bit ARM), and Arduino Due (84 MHz 32-bit ARM). Among them, Arduino Uno has computationally weak hardware, which helps to verify the performance on resource-constrained devices. Figure 4 demonstrates the diagram of our experimental setup.

4.2 Security Threat Analysis

General Security Requirements
First, we analyze that LaaCan satisfies confidentiality, integrity, and authenticity. For the purpose of analysis, we name the ECUs as ECU_T (transmitter), ECU_R (receiver), and ECU_A (attacker). We assume ECU_T and ECU_R are legitimate nodes and ECU_A is an attacker for the following scenarios.

Confidentiality. ECU_T sends data to ECU_R, and ECU_A tries to sniff the data. We observe that before installing LaaCan, ECU_A receives the original data. After applying LaaCan, ECU_A receives the encrypted data. Encrypted data is completely different from the original data and meaningless to the attacker node.

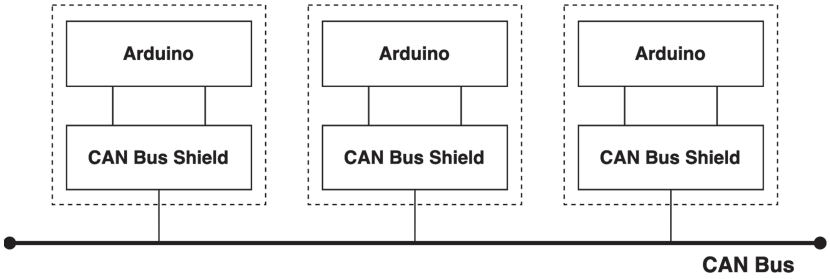

Fig. 4. The experimental setup diagram.

Integrity. ECU_T tries to send data after encryption. However, one or more bit(s) of transmitted data changes due to transmission error or attacker's manipulation and ECU_R receives the fabricated data. We observe that ECU_R does not process the altered data because of the mismatch in the MAC.

Authenticity. Similar to integrity, MAC also ensures data authenticity. ECU_T sends data to ECU_R. However, the message identifier got altered due to transmission error or attacker's manipulation. ECU_R generates the MAC with respect to the altered identifier and discards the message because of mismatch in MAC.

Some Attack Defenses

We analyze the defense potential of LaaCan against five security attacks: eavesdropping, spoofing, replay, Man-in-the-Middle (MITM), and remote attacks. These attacks are considered for protection analysis as they are the most common form of attacks in CAN protocol [4,29]. Denial of Service (DoS) attacks are not considered as preventing them in CAN protocol is impossible [31] and the existing solutions also struggle against it [26].

Eavesdropping Attacks. In a broadcast network, all the participant nodes receive all the transmitted packets. Hence, an adversary ECU can read the data transmission to retrieve information, which may lead to future attacks. Data transmission is encrypted in LaaCan. Therefore, eavesdropping attacks cannot compromise data.

Spoofing Attacks. Spoofing attacks are done by impersonating another ECU. An adversary can mislead a vehicle subsystem by performing this kind of attack. These attacks are possible if the network lacks authenticity and integrity. LaaCan ensures the authenticity and integrity of the transmitted messages to protect against spoofing attacks.

Replay Attacks. In a replay attack, a copy of a previously recorded valid message is transmitted. As messages are exchanged in broadcast form, any attacker node connected to the network can perform a replay attack if no security measures are not taken [19]. The receiving node cannot differentiate between a valid and

replayed message when message freshness is not ensured. LaaCan eliminates replay attacks by involving a message counter in the keystream derivation process that assures message freshness.

Man-in-the-Middle (MITM) Attacks. MITM attacks control the communication between two legitimate nodes secretly. Buttigieg et al. [12] demonstrated a MITM attack by installing an attacker node between an instrument cluster and a vehicle simulator. In LaaCan, the attacker node cannot retrieve the actual information from the encrypted message. Also, the possible altered relayed message cannot bypass the authentication process. Hence, LaaCan prevents MITM attacks.

Remote Attacks. Modern vehicles have an OBD-II port that provides self-diagnostic and reporting support. OBD-II is connected to in-vehicle network. Woo et al. [33] presented a remote attack that connects a malicious mobile application to a third party OBD-II diagnostic tool. These attacks are possible due to the lack of data authenticity. LaaCan provides authenticity to prevent these attacks. The secret-key and cryptography mechanism have to be shared with trusted manufacturers of OBD-II dongles to make them compatible with the proposed architecture. If a dongle is vulnerable and a secret key is compromised, then the attacker may bypass the security measures. However, it applies to all the cryptographic approaches to some extent as they operate based on the secret key.

4.3 Evaluation Metrics and Comparison

We evaluate LaaCan based on four metrics: communication overhead, bus load, deployment cost, and compatibility. The metrics are discussed in next paragraphs. Afterwards, it is compared with some existing solutions based on security requirements and performance.

Evaluation Metrics
Communication Overhead. CAN is meant for real-time communication. Therefore, communication overhead is the most important evaluation criteria. LaaCan involves encryption and authentication, which requires processing of every message. It has a total delay of 1.52 ms 16 MHz configuration, 0.25 ms in 48 MHz, and 0.07 ms in 84 MHz. We observe that communication delay significantly depends on the hardware configuration of ECUs. All the delays are for 8 bytes of data, which is the maximum payload of the protocol. We do not include transmission delay here as it solely depends on the setup environment.

Bus Load. If security measures increase the load of the bus, it must have an impact on the message processing latency. Woo et al. [33] use an extra message to transmit MAC for each message, which at least double the traffic of the network. LaaCan avoids transmission of additional messages by replacing the CRC field with the MAC. Hence, LaaCan does not have any effect on the bus load.

Deployment Cost. A huge number of already manufactured vehicles are using CAN protocol. Hence, a new security solution must have a low installation cost. Some of the existing solutions [9,25] install one or more new dedicated ECUs for security purposes, which is costly. LaaCan has a low deployment cost as it can be integrated by software updates only.

Compatibility.[1] LaaCan integration does not have any impact on the core CAN protocol. No additional bits are added to the message. LaaCan replaces the CRC field with MAC to avoid use of CAN+ [35] or any other customized version of the protocol. However, error checking is part of integrity process and LaaCan assures integrity. Hence, the modification follows the core specification of CAN.

Comparative Analysis

Table 3 illustrates the security features and performance comparison of the proposed design with some related work discussed in Sect. 2. Since LaaCan adapts the cryptographic architecture, IDS-based solutions are excluded from the comparison.

Table 3. Security and performance comparison.

Solution	Confidentiality	Integrity	Authenticity	Bus load	Deployment cost	Compatibility	Overhead
CANTrack [14]	✓	×	×	Unchanged	Low	High	-
VeCure [32]	×	✓	✓	High	Low	High	0.05 ms (40 MHz)
CANAuth [31]	×	✓	✓	Unchanged	Low	Low	-
LiBrA-CAN [15]	×	✓	✓	High	High	Low	2.54 ms
WooCAN [33]	✓	✓	✓	Medium	Low	High	0.38 ms (60 MHz)
LASAN [22]	✓	✓	✓	Medium	Very High	High	4.6 ms (168 MHz)
TOUCAN [8]	✓	✓	✓	Unchanged	Low	Very Low	2.35 ms (168 MHz)
CaCAN [25]	×	✓	✓	Medium	High	Low	0.03 ms (50 MHz)
TCAN [9]	×	×	✓	Medium	Very High	Low	0.03 ms
LaaCan	✓	✓	✓	Unchanged	Low	High	1.52 ms (16 MHz) 0.25 ms (48 MHz) 0.07 ms (84 MHz)

CANTrack [14] only encrypts data to ensure confidentiality. Along with LaaCan, there are three solutions [8,22,33] that ensure confidentiality, integrity, and authenticity. The other solutions implement an authentication mechanism to ensure integrity and authenticity.

The additional message transmission needed for security purposes increases the traffic of the network. CANTrack [14], TOUCAN [8], VeCure [32], and LaaCan have unchanged network traffic. As CANTrack does not authenticate the message, it does not need to transmit MAC. TOUCAN reduces the payload size to append MAC to it. VeCure and LaaCan replace the CRC field with MAC to avoid additional message transmission.

[1] Compatibility indicates the degree of change required in standard CAN protocol and it is a subjective metric.

LASAN [22] and TCAN [9] have very high deployment costs as multiple dedicated ECUs are required to be installed for security. LiBrA-CAN [15] and CaCAN [25] require the installation of one dedicated ECU. Also, the dedicated ECUs used for security purposes increase the coupling among them and general ECUs, which is considered as a weakness [21]. The other considered solutions including LaaCan, have low deployment cost as no hardware changes are required.

The highly compatible solutions do not change the specifications of the standard protocol and do not involve a different variant of the protocol. TOUCAN [8] has very low compatibility as it reduces the payload size. Also, CANAuth [31] and LiBrA-CAN [15] have low compatibility as they use CAN+ [35] that requires hardware changes in transceiver [8]. Though we replace the CRC field with MAC, LaaCan can run on existing hardware and network configurations without any modification. Therefore, LaaCan is highly compatible with standard CAN.

A comparative analysis with other work in terms of communication overhead is challenging as the overhead depends on hardware configurations. Since the clock speed of the processor is considered as one of the significant factors behind the performance, we perform the analysis based on the clock speed. VeCure [32], CaCAN [25], and TCAN [9] outperform LaaCan in terms of overhead. However, they fail to assure at least one feature of confidentiality, integrity, and authenticity. Also, CaCAN and TCAN have high deployment cost, and low compatibility with the standard CAN protocol. Though WooCAN [33], LASAN [22], and TOUCAN [8] ensure the three security features, they have significantly high communication delay than LaaCan.

The remote attacks are usually initiated through the In-Vehicle Infotainment (IVI) system and OBD-II. The secret key and cryptographic mechanism have to be shared with these systems to make them compatible with LaaCan. However, if these systems do not have protected memory and compromise the secret key, then adversaries can bypass the authentication system. VeCure [32] attempts to mitigate this issue by sharing the secret key only with the high-trust ECUs that do not have any external interfaces. It only authenticates the communications between high-trust ECUs and does not assure confidentiality, which makes the ECUs with external interfaces such as IVI and OBD-II vulnerable. Any non-trusted OBD-II tool can read and learn from the data as there is no encryption in place. Also, it can send malicious messages to IVI that will not be discarded as there is no authentication done in IVI.

5 Conclusion and Future Work

5.1 Conclusion

The automotive industry is advancing towards the adoption of information technology and electronic components. The involvement of information technology has opened up the in-vehicle communication networks to the cyber world. Adversaries can gain access to an in-vehicle network by exploiting the vulnerabilities

of the Controller Area Network (CAN). In this work, we design a lightweight authentication architecture called LaaCan to secure CAN network communication. We classify the existing security solutions for CAN on the basis of security enforcement procedures in order to compare LaaCan with the existing work. LaaCan is an Authenticated Encryption with Additional Data (AEAD)-based security architecture that implements ChaCha20-Poly1305 for one pass encryption-authentication process. It protects the network from eavesdropping, spoofing, replay, Man-in-the-Middle (MITM), and remote attacks by ensuring the integrity, authenticity, and confidentiality of the transmitted data. The experimental results illustrate that the communication delay can be reduced to 0.07 ms. We evaluate LaaCan based on the communication delay, traffic load, deployment cost, and compatibility with the standard protocol. The comparative analysis shows that the proposed architecture suffers from less overhead compared to the solutions with similar security measures. LaaCan does not increase network traffic. A software update can incorporate the solution without any hardware changes, and it has high compatibility. Lastly, LaaCan is compliant with the CAN and AUTOSAR standards.

5.2 Limitations and Future Work

The cryptographic algorithm requires a secret key, message counter, session id, and session key. These values have to be stored, which consume memory storage. Besides, we generate a session key from the pre-shared secret key. If an adversary compromises a session key, it can bypass the security measures for that particular session. While compromising the session key requires the knowledge of the pre-shared secret key and session id, the attacker may compromise the encryption algorithm that generates the key. We assume that these values are stored in more protected memory. LaaCan uses a message counter to assure message freshness. However, we do not transmit freshness value due to the limited size of a CAN message. Since CAN is a broadcast-based network, all the ECUs receive the transmitted messages in network. Thus, the counter increments upon receiving every message, which keeps all the ECUs synchronized. If an ECU somehow does not receive some messages, then it cannot synchronize with the network.

If the pre-shared key is compromised, an adversary can create a legitimate session key. A sophisticated secret key generation and sharing mechanism can address the issue so that an adversary cannot create a legitimate session key from a compromised pre-shared key. However, the generation of secret keys requires a lot of processing power. As a result, we plan to implement a secret key sharing mechanism for low powered ECUs. Also, we conduct our experiments with microcontrollers that are not real ECUs. Therefore, we plan to install our solutions in ECUs with a real vehicle system running. It will help us reach more concrete conclusions concerning the actual performance and feasibility.

Acknowledgments. This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) program. The authors would also like to thank Farnood Faghihi for reviewing the paper.

References

1. AUTOSAR - enabling innovation. https://www.autosar.org/. Accessed 12 Apr 2020
2. Iso 11898–1:2003 - road vehicles - Controller Area Network (CAN) - part 1: Data link layer and physical signalling. https://www.iso.org/standard/33422.html. Accessed 12 Apr 2020
3. What is AUTOSAR and why is it important. https://www.fpt-software.com/automotive-tech-blog/what-is-autosar-and-why-is-it-important/. Accessed 12 Apr 2020
4. AbdAllah, E.G., Zulkernine, M., Gu, Y.X., Liem, C.: Towards defending connected vehicles against attacks. In: Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems, pp. 1–9 (2017)
5. Alam, M.S.U., Iqbal, S., Zulkernine, M., Liem, C.: Securing vehicle ECU communications and stored data. In: ICC 2019–2019 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2019)
6. AUTOSAR: Specification of secure onboard communication. https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_SecureOnboardCommunication.pdf
7. Avatefipour, O., Malik, H.: State-of-the-art survey on in-vehicle network communication (CAN-Bus) security and vulnerabilities. Int. J. Comput. Sci. Netw. **6** (2017)
8. Bella, G., Biondi, P., Costantino, G., Matteucci, I.: Toucan: a protocol to secure controller area network (CAN). In: Proceedings of the ACM Workshop on Automotive Cybersecurity. AutoSec 2019, pp. 3–8. ACM, New York (2019)
9. Biham, E., Bitan, S., Gavril, E.: TCAN: authentication without cryptography on a CAN bus based on nodes location on the bus. In: 2018 Embedded Security in Cars, November 2018
10. Bittl, S.: Attack potential and efficient security enhancement of automotive bus networks using short macs with rapid key change. In: Sikora, A., Berbineau, M., Vinel, A., Jonsson, M., Pirovano, A., Aguado, M. (eds.) Communication Technologies for Vehicles. Nets4Cars/Nets4Trains/Nets4Aircraft 2014. LNCS, vol. 8435, pp. 113–125. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06644-8_11
11. Boldt, B.: Automotive security in a CAN. https://www.electronicdesign.com/markets/automotive/article/21805532/automotive-security-in-a-can. Accessed 07 May 2020
12. Buttigieg, R., Farrugia, M., Meli, C.: Security issues in controller area networks in automobiles. In: 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), pp. 93–98, December 2017
13. De Santis, F., Schauer, A., Sigl, G.: ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2017, pp. 692–697, March 2017
14. Farag, W.A.: Cantrack: enhancing automotive CAN bus security using intuitive encryption algorithms. In: 2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), pp. 1–5. IEEE (2017)

15. Groza, B., Murvay, S., van Herrewege, A., Verbauwhede, I.: LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks. In: Pieprzyk, J., Sadeghi, A.-R., Manulis, M. (eds.) CANS 2012. LNCS, vol. 7712, pp. 185–200. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35404-5_15
16. Hu, Q., Luo, F.: Review of secure communication approaches for in-vehicle network. Int. J. Automot. Technol. **19**(5), 879–894 (2018)
17. SAE International: Serial control and communications heavy duty vehicle network (2019)
18. Keller, J.: Best third-party Carplay apps. https://www.imore.com/best-third-party-carplay-apps (2018). Accessed 24 Dec 2019
19. Lin, C., Sangiovanni-Vincentelli, A.: Cyber-security for the controller area network (CAN) communication protocol. In: 2012 International Conference on Cyber Security, pp. 1–7, December 2012
20. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In: Joux, A., Youssef, A. (eds.) SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13051-4_19
21. Moukahal, L., Zulkernine, M.: Security vulnerability metrics for connected vehicles. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 17–23. IEEE (2019)
22. Mundhenk, P., et al.: Security in automotive networks: lightweight authentication and authorization. ACM Trans. Des. Autom. Electron. Syst. (TODAES) **22**(2), 25 (2017)
23. NG, C.: What is data integrity and how can you maintain it? https://www.varonis.com/blog/data-integrity/. Accessed 07 May 2020
24. Nir, Y.: ChaCha20 and poly1305 for IETF protocols. https://tools.ietf.org/html/rfc7539
25. R. Kurachi, Y. Matsubara, H.T.N.A.Y.M., Horihata, S.: CaCAN - centralized authentication system in CAN. In: 2014 Embedded Security in Cars, November 2014
26. Radu, A.-I., Garcia, F.D.: LeiA: a lightweight authentication protocol for CAN. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9879, pp. 283–300. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45741-3_15
27. Silva, N.B., Pigatto, D.F., Martins, P.S., Branco, K.R.: Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer. J. Netw. Comput. Appl. **60**, 130–143 (2016)
28. Studio, S.: CAN-Bus shield v2.0. http://wiki.seeedstudio.com/CAN-BUS_Shield_V2.0/
29. Sun, J., Iqbal, S., Seifollahpour Arabi, N., Zulkernine, M.: A classification of attacks to in-vehicle components (IVCs). Veh. Commun. **25**, 100253 (2020)
30. Ujiie, Y., et al.: A method for disabling malicious can messages by using a centralized monitoring and interceptor ECU. In: 2015 Embedded Security in Cars (2015)
31. Van Herrewege, A., Singelee, D., Verbauwhede, I.: CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus. In: ECRYPT Workshop on Lightweight Cryptography, vol. 2011 (2011)
32. Wang, Q., Sawhney, S.: VeCure: a practical security framework to protect the CAN bus of vehicles. In: 2014 International Conference on the Internet of Things (IOT), pp. 13–18. IEEE (2014)

33. Woo, S., Jo, H.J., Lee, D.H.: A practical wireless attack on the connected car and security protocol for in-vehicle CAN. IEEE Trans. Intell. Transp. Syst. **16**(2), 993–1006 (2015)
34. Yang, J., et al.: Data management for automotive ECUs based on hybrid RAM-NVM main memory. In: 2016 13th International Conference on Embedded Software and Systems (ICESS), pp. 74–79 (2016)
35. Ziermann, T., Wildermann, S., Teich, J.: CAN+: a new backward-compatible controller area network (CAN) protocol with up to 16x higher data rates. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 1088–1093. European Design and Automation Association (2009)