WIP: The Feasibility of High-Performance Message Authentication in Automotive Ethernet Networks

Evan Allen
Virginia Tech
evanallen@vt.edu

Zeb Bowden
Virginia Tech
Transportation Institute
zbowden@vt.edu

Randy Marchany
Virginia Tech
marchany@vt.edu

J. Scot Ransbottom
Virginia Tech
ransbottom@vt.edu

***Abstract*—Modern vehicles are increasingly connected systems that expose a wide variety of security risks to their users. Message authentication prevents entire classes of these attacks, such as message spoofing and electronic control unit impersonation, but current in-vehicle networks do not include message authentication features. Latency and throughput requirements for vehicle traffic can be very stringent ($<$0.1 ms and $>$100 Mbps in cases), making it difficult to implement message authentication with cryptography due to the overheads required. This work investigates the feasibility of implementing cryptography-based message authentication in Automotive Ethernet networks that is fast enough to comply with these performance requirements. We find that it is infeasible to include Message Authentication Codes in all traffic without costly hardware accelerators and propose an alternate approach for future research to minimize the cost of authenticated traffic.**

I. Introduction

In recent decades, modern 'smart' vehicles have become increasingly connected systems that expose serious security risks to their users.

New vehicles contain numerous inter-connected electronic control units (ECUs) that handle various aspects of the car's operation, such as the anti-lock braking system, powertrain control, speedometer control, and more [11]. These ECUs expose a variety of attack surfaces, such as physical access through the Onboard Diagnostics II (OBD-II) diagnostic port, close proximity access through Bluetooth/Wi-Fi communications, or even remote access through radio or cellular (3/4/5G) means [17]. It is difficult to ensure an attacker cannot enter an in-vehicle network (IVN) by any of these means.

If attackers do gain access to an IVN, they can cause serious damage. For example, Miller and Valasek were able to affect the steering and braking systems of Fiat-Chrysler vehicles remotely over the Sprint cellular network, causing the recall of 1.4 million vehicles [15]. A malicious user could abuse this to cause a vehicle to suddenly stop, drive somewhere else, or even crash.

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23035
www.ndss-symposium.org

A. Message Authentication

One way to make IVNs more robust is to implement message authentication, which allows receiver ECUs to verify that a message actually came from a trusted sender ECU. This would prevent entire classes of attacks, such as ECU impersonation and message injection [17].

Most widely adopted IVN technologies do not implement message authentication. The Controller Area Network (CAN) uses a bus topology where all ECUs can send messages to all ECUs without identifying the sender, making it trivial to spoof messages from other ECUs [19]. FlexRay is similarly vulnerable to message spoofing attacks [20], and Automotive Ethernet (AE) does not have any widely adopted authentication mechanism either [7].

Message authentication is commonly implemented in conventional information technology systems using Message Authentication Codes (MACs), which are cryptographic tags a sender appends to a message that verify the message's authenticity and integrity. Multiple works have attempted to implement MACs for message authentication in IVNs (see Section II), but many struggle to meet the modern performance requirements of automotive traffic (see Section I-B).

B. Performance Concerns

IVNs contain many classes of data with different performance requirements, outlined in Table I from [3], [13], [23]. Some of these data classes require ultra-low latencies (such as 0.1 ms for critical control data), while others require very high throughputs (such as up to 100 Mbps for lidar data). These requirements can be difficult to meet in the resource-constrained environment of a vehicle.

TABLE I. Performance requirements for various classes of data, adapted from [3], [13], [23]

Data Class	Throughput (Mbps)	Max. Latency (ms)	Period (ms)
Critical control	0.5-1	0.1	Event driven
Normal control	0.5-1	5-50	5-50
Radar	0.1-15	10	10
Ultrasonic	0.01-0.23	20	20
Camera Video*	$\sim$52	33	33
Lidar	20-100	10	10

*30 frames per second, compressed

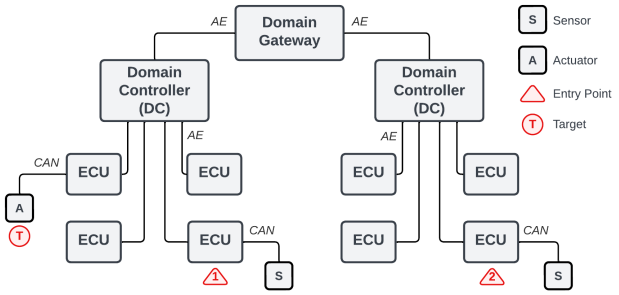

Fig. 1. The threat model, consisting of a domain-controller architecture [6] with ECUs and domain controllers connected using Automotive Ethernet (AE). An attacker wishes to send a malicious message to a target actuator T from a compromised ECU either (1) in the same domain or (2) in another domain. A similar model could be adopted for a zonal architecture as well.

C. Automotive Ethernet (AE)

This work focuses on message authentication in the context of an AE [9] network. AE is a promising new IVN technology that provides very high throughput (up to 10 Gbps) at low, bounded latencies using Time-Sensitive Networking (TSN) standards [1], [22]. AE networks primarily differ from conventional Ethernet networks in their physical layer, which consist of twisted-pair wires to satisfy vehicle requirements for resistance to electromagnetic interference [3].

The high throughput of AE networks is important in modern vehicles because (1) sensors such as lidar and cameras send data at higher speeds than CAN, FlexRay, or other IVN technologies can support [3]; and (2) car manufacturers are moving towards a 'zonal' architecture consisting of a few powerful ECUs connected by a very high-speed (Gbps) backbone [6]. AE can achieve bounded low latency using standards created by the IEEE 802 Time-Sensitive Networking Task Force [1], which include provisions for both scheduled and event-driven, real-time traffic.

D. Threat Model - Spoofing Messages

This paper assumes a domain-based architecture [6] as shown in Figure 1 where an attacker wishes to send a malicious message M to a target actuator T from a compromised ECU either in (1) the same domain or (2) a different domain. The message M is one *that the compromised ECU should not be able to send*, and is instead *spoofed* as if it came from another ECU.

For example, the attacker may compromise a speed sensor ECU and try to send `brake` packets to the Anti-Lock Bracking System (ABS) ECU as if they came from a higher-level control ECU. We investigate how authentication could be implemented in IVNs to prevent such message spoofing attacks.

In this paper, we do not focus on attacks within an ECU's local CAN network. The threat model could be adapted to the zonal architecture by changing the domain controllers and gateway to their zonal counterparts.

E. Contributions and Structure

This paper reviews existing work in the AE authentication space in Section II. It analyzes the latency components of one existing AE security protocol in Section III and then investigates ways to increase throughput of MAC generation and verification in Section IV. The paper discusses the results and proposes a possible solution in Section V, and then concludes in Section VI.

II. Existing Work

There already exist multiple prior works in the AE authentication space. Corbett et al. [4] suggested using MACs for lightweight authentication in AE applications in 2016. Li et al. [14] proposed an improved authentication scheme for IVNs using a faster version of AES-128. Wang and Ganesan [24] discuss using MACs for message authentication in CAN and AE and compare the merits of hardware- vs. software-based cryptography for MAC usage.

Ju et al. [10] emphasize the reality that vehicles are made of varying types of networks (including CAN, FlexRay, MOST, and LIN subnetworks) with varying performance and security requirements. They suggest assigning data to separate 'security levels' similar to the concept of bandwidth classes. Certain security levels would include authentication, while others would not.

A. MACsec [8]

MACsec [8] is a recent standard for Ethernet designed to provide security at the link layer between connected nodes. It provides services such as data confidentiality, message authentication, and integrity checking.

Pena et al. [18] evaluated the impact of MACsec on the performance of traffic traveling through a set of TSN switches. Encouragingly, they found that MACsec did not create large performance overheads when implemented in hardware.

1) Latency: Their original setup with plain Ethernet on TSN switches achieved latencies between 8,698 ns and 55,314 ns for various frame sizes, which is under the 0.1 ms latency threshold in Table I. Introducing MACsec changed these values by less than 1,000 ns. However, they did not test the latency of software implementations of MACsec.

2) Throughput: As before, Pena et al. achieved high throughput speeds of over 200 Mbps with both plain Ethernet and hardware-accelerated MACsec. These are sufficient for Table I. However, they found that their throughput dropped by 44% when they attempted to use the Linux Kernel's software implementation of MACsec instead. It is worth noting that the software performance can vary greatly; the authors of [7] found only 2.87 Mbps when benchmarking MACsec in software.

B. Gatekeeper [7]

Another recent proposal for a fast AE security protocol with message authentication was Gatekeeper [7]. Gatekeeper uses a star topology where all messages pass through a central, on-path 'authenticator' node that verifies their authenticity before passing them to receiver ECUs. All nodes have separate secret keys that they individually share with the authenticator, allowing them to send and receive MACs from the authenticator using high-speed symmetric cryptography.

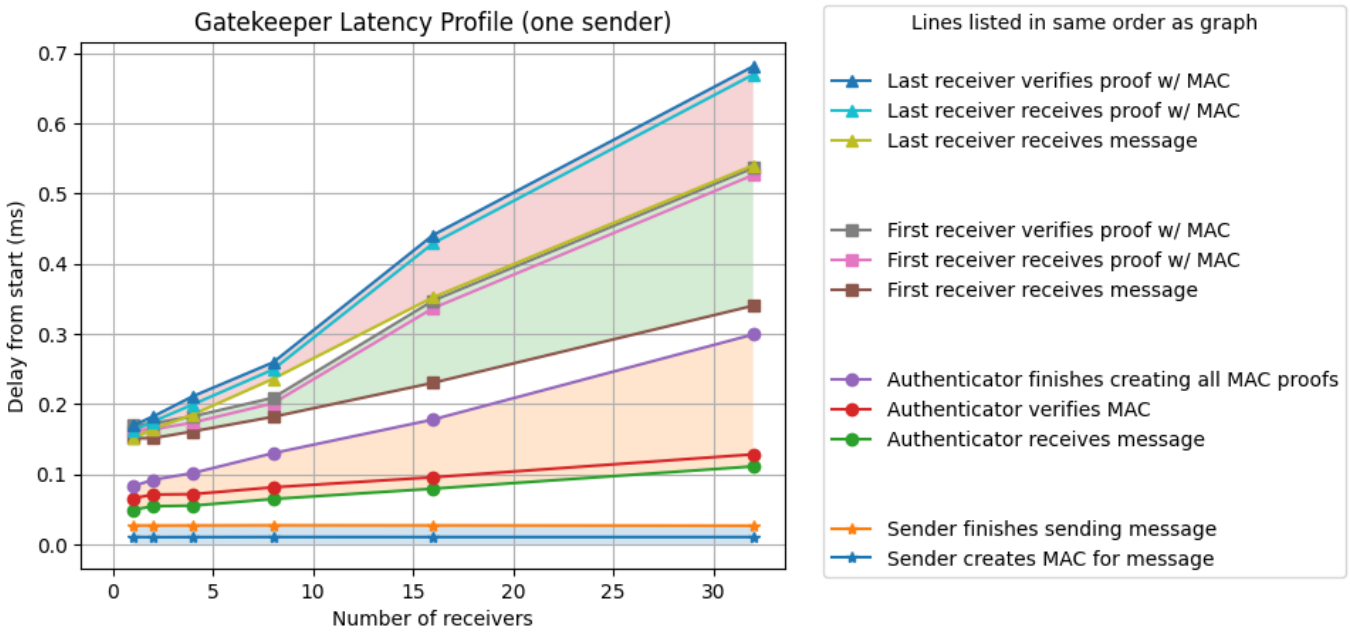

Fig. 2. Latency profile for Gatekeeper, simulated with one sender transmitting CAN over UDP to varying numbers of receiver ECUs. Each line represents the time it took for a packet to finish that step during transmission, averaged over 256 different measurements. Steps are grouped together in four main groups by color highlight (from bottom to top: sender steps (blue), authenticator steps (orange), first receiver steps (green), last receiver steps (pink)). White areas represent time packets are in transit. The senders, receivers, and authenticator ECUs were all simulated using a Docker-based software testbed as in the original paper.

To send a message, a sender ECU would generate a MAC using its secret key and send it to the authenticator. The authenticator would verify the authenticity of the message by checking the MAC, and then would forward the message to all receiver nodes with an additional 'proof packet' for each that contained a new MAC using each receiver's secret key. The receiver could thus be confident in a message's authenticity if it received a correct proof packet from the authenticator.

While faster than most previous work, Gatekeeper's authors found significant difficulty in meeting the latency and throughput requirements shown in Table I. They conducted all their tests in a software, Docker-based testbed.

1) Latency: In the original paper, Gatekeeper was only able to achieve the critical control latency requirement of 0.1 ms when there was only a single sender ECU and a single receiver ECU with no additional traffic. Any additional receiver ECUs or traffic caused additional latency that violated the requirement. The authors argue much of this latency is due to their software testbed.

2) Throughput: Gatekeeper's authors found that they could not satisfy the throughput requirements in Table I because the cryptographic/hashing functions they used were too slow. They ran benchmarks of AES-128-CBC, AES-128-CCM, AES-128-GCM, SHA256, HMAC-SHA256, and ChaCha20-Poly1305 on an automotive development board and found that the highest throughput they could achieve through any of them was about 14 Mbps with ChaCha20-Poly1305. This is much lower than the worst-case lidar throughput requirements ($\sim$100 Mbps) and certainly would be too slow for the Gbps-speed backbone of a zonal architecture.

They did find that they could achieve higher speeds with hardware-accelerated AES-128-GCM, further indicating that hardware accelerated cryptography might provide adequate performance.

III. Latency Analysis of Gatekeeper

To understand how much time MAC operations require and why Gatekeeper struggles to meet the most stringent 0.1 ms latency requirement in software, we profiled the original Gatekeeper implementation to see how long each part takes.

We attempted to replicate the original Gatekeeper testing environment as much as possible using the Docker-based testbed from the paper. We limited the performance of the sender and receiver ECU containers to simulate a resource-constrained environment but left the central authenticator unconstrained as in the original paper.

Figure 2 shows the time taken by various steps of the Gatekeeper process for packets sent from a single sender ECU to a varying number of receiver ECUs. As in the original paper, end-to-end latency increases linearly with receiver count. None of our trials satisfied the most stringent 0.1 ms deadline from Table I.

The graph corroborates the Gatekeeper authors' claim that their implementation was slowed by their software testbed and the packet switching between their containers, though. In the plot, most of the receivers' increased delay is due to waiting to receive a packet in transit. Notice how small the receiver verification time is compared to the time from message reception to proof reception. Most MAC generation and verification took approximately 0.01-0.02 ms.

However, the end-to-end latency of Gatekeeper excluding packet transit time (white space) is still too high ($>$0.1 ms) for even one receiver and increases for more of them.

IV. Achieving High Throughput

There are two main ways to increase throughput of message authentication cryptography: speed up the algorithm (lightweight cryptography) or speed up the computation itself (hardware acceleration).

A. Lightweight Cryptography

There exist many lightweight MAC schemes designed for resource-constrained environments, including Chaskey [16], LightMAC [21], and SipHash [2]. While faster than conventional algorithms in some cases (such as Chaskey beating AES-CBC-MAC and Poly1305 for short messages in [5]), they still struggle to approach the ∼80+ Mbps speeds modern sensors and backbone connections need in automotive applications.

B. Hardware Acceleration

Hardware-based cryptography accelerators have shown promise in providing fast enough MAC generation and verification implementations. The authors of [7] noticed speedups from 3.24 Mbps to 81.42 Mbps in AES-128-GCM encryption when switching to a hardware security module, and Kieu-Do-Nguyen et al. [12] achieved 4.8 Gbps throughput on a new hardware implementation of HMAC-SHA2 using a Field-Programmable Gate Array (FPGA). We also previously discussed how Pena et al. showed the feasibility of running MACsec at high speeds with hardware in Section II.

However, hardware security modules providing this functionality can quickly become expensive and are unrealistic to deploy on every ECU.

V. Discussion and Future Work

Our analysis of the Gatekeeper protocol and other existing work in this space indicates that it is not feasible to implement message authentication at the ultra-low latencies (<0.1 ms) and high throughputs (>80 Mbps) of Table I without significant extra computational resources such as external hardware cryptography accelerators, which would be prohibitively expensive if used in all ECUs.

As with all engineering tradeoffs, this indicates that a nuanced approach is necessary. Existing software methods such as Gatekeeper are still useful in situations that do not require the most stringent performance requirements, such as ultrasonic data or door control. But for classes like critical control data, a better solution might be to reduce the need for cryptographic authentication at all.

A. Port-based Network Access Control/Firewall

One idea is to use hardware ports to authenticate packets in switched Ethernet networks. Consider again the network in Figure 1. Because ECUs do not share a common bus,

1) ECUs can trust that all incoming messages are from the domain controller, effectively offloading their trust and authentication burden to that domain controller, and
2) The domain controller can trust that all incoming messages on a certain hardware interface are from the ECU assigned to that interface (even if that ECU is compromised).

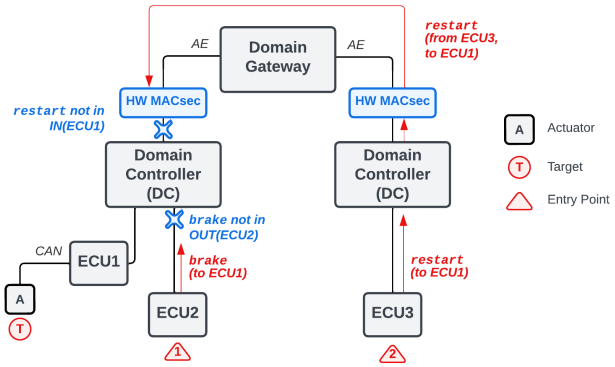

Fig. 3. Proposed architecture. ECUs trust traffic coming from their domain controllers (DCs), and DCs authenticate child ECUs based on their hardware interface. DCs authenticate inter-domain traffic using hardware (HW) MACsec. In this example, an attacker compromises ECU2 and ECU3 and tries to send ECU1 malicious messages. However, ECU2 is not allowed to send the attempted message and ECU1 is not allowed to receive the attempted message from ECU3, so the attack fails.

This means that the domain controller would get the opportunity to verify the authenticity of all traffic from its child ECUs, and all of its child ECUs could rely on the domain controller to authenticate their traffic for them. The only need for cryptographic authentication would be at the inter-domain level, since a domain controller cannot associate a single ECU to its inter-domain link like it can for its child ECU links. This is tolerable since is much more feasible to implement hardware cryptography (such as MACsec) for a few domain controllers than all ECUs.

With this setup, all inter-ECU messages are authenticated, allowing the domain controllers to detect and stop message spoofing by enforcing a security policy about what types of packets each ECU should be able to send and receive.

For example, consider the set of ECUs ($\mathbb{E}$) and possible message types ($\mathbb{M}$). $\mathbb{M}$ could consist of message types such as `brake`, `speed_reading`, or more. For each ECU $E \in \mathbb{E}$, there would be a list of message types allowed in / out:

$$IN(E) \subseteq \mathbb{M} \quad (1)$$

$$OUT(E) \subseteq \mathbb{M} \quad (2)$$

Where $IN(E)$ and $OUT(E)$ are as restrictive as possible.

With this policy and the ability for domain controllers to authenticate their child ECU's messages based on the interface they are connected on, domain controllers could block forbidden outbound messages from the ECUs and block forbidden inbound messages to ECUs. This would prevent an attacker from using a compromised ECU to spoof messages from other ECUs (see Figure 3).

This approach would remove the need for most ECUs to do extra cryptographic work. It does not protect against attackers with physical access to the vehicle, but such attacks are much less likely than remote or near-field attacks such as Miller and Valasek's [15]. It also does not consider authentication of

devices on internal CAN networks behind each ECU, but that is not our focus in this work.

For future work, we plan to flesh out the domain controller firewall approach above and build a prototype to investigate the performance and limitations of such a system. Can it satisfy the requirements of Table I? How restrictive can the IN / OUT policies be? How much overhead do they cause? Finding useful results here could provide better insight into the new possibilities opened by switched Ethernet networks in vehicles, especially as manufacturers begin to shift into domain controller and zonal architectures [6].

We also plan to try to reproduce the MACsec benchmark results in [18] and determine if MACsec can remain performance-compliant on low-cost hardware. This would make it possible for automotive manufacturers to begin including security protocols like MACsec at a practical cost in their vehicles.

VI. Conclusion

This work investigated the need for high-performance message authentication in vehicles and how feasible it might be to implement in AE networks.

We found that performance requirements for in-vehicle traffic range dramatically in both latency and throughput, demanding at times sub-0.1-ms latency and up to 100 Mbps throughput for a single sensor. This paper reviewed previous work in the field and focused on Gatekeeper, a recent proposal for an AE security protocol that highlights the difficulties in meeting these latency and throughput requirements.

We then profiled Gatekeeper and investigated other lightweight MAC algorithms to find that it is not feasible to meet the most stringent latency and throughput requirements without costly resources such as hardware cryptography accelerators.

Finally, we proposed a new approach to designing IVNs to reduce the amount of cryptographic authentication required.

References

[1] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, and Saad Mubeen. Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. *Journal of Systems Architecture*, 117:102137, August 2021.

[2] Jean-Philippe Aumasson and Daniel J. Bernstein. SipHash: A Fast Short-Input PRF. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Steven Galbraith, and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012*, volume 7668, pages 489–508. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[3] Eunmin Choi, Hoseung Song, Suwon Kang, and Ji-Woong Choi. High-Speed, Low-Latency In-Vehicle Network Based on the Bus Topology for Autonomous Vehicles: Automotive Networking and Applications. *IEEE Vehicular Technology Magazine*, 17(1):74–84, March 2022.

[4] Christopher Corbett, Elmar Schoch, Frank Kargl, and Felix Preussner. Automotive Ethernet: security opportunity or challenge? In Michael Meier, Delphine Reinhardt, and Steffen Wendzel, editors, *Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit*, pages 45–54, Bonn, 2016. Gesellschaft für Informatik e.V.

[5] Sébastien Duval and Gaëtan Leurent. Lightweight MACs from Universal Hash Functions. In Sonia Belaïd and Tim Güneysu, editors, *Smart Card Research and Advanced Applications*, volume 11833, pages 195–215. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[6] GuardKnox. Automotive Zonal Architecture, 2022.

[7] Shengtuo Hu, Qingzhao Zhang, André Weimerskirch, and Z. Morley Mao. Gatekeeper: A Gateway-based Broadcast Authentication Protocol for the In-Vehicle Ethernet. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, pages 494–507, New York, NY, USA, May 2022. Association for Computing Machinery.

[8] IEEE. IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security. Technical Report 802.1AE-2018, IEEE, December 2018. ISBN: 9781504452151.

[9] IEEE. *802.3-2022 - IEEE Standard for Ethernet*. IEEE, 2022. OCLC: 1343075176.

[10] Hongil Ju, BooSun Jeon, Daewon Kim, Boheung Jung, and Kyudong Jung. Security Considerations for In-Vehicle Secure Communication. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1404–1406, Jeju Island, Korea (South), October 2019. IEEE.

[11] Narayan Khatri, Rakesh Shrestha, and Seung Yeob Nam. Security Issues with In-Vehicle Networks, and Enhanced Countermeasures Based on Blockchain. *Electronics*, 10(8):893, April 2021.

[12] Binh Kieu-Do-Nguyen, Trong-Thuc Hoang, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham. High-performance Multi-function HMAC-SHA2 FPGA Implementation. In *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, pages 30–34, Quebec City, QC, Canada, June 2022. IEEE.

[13] Yong Kim and Masa Nakamura. Automotive Ethernet Network Requirements, March 2011.

[14] Jia-Ming Li, Yu-Jing Wu Shuo-Fu, and Yi-Nan Xu. High-Efficiency Encryption and Authentication Network Security for Automotive Ethernet. *International Journal of Modeling and Optimization*, 12(2), 2022.

[15] Charlie Miller and Chris Valasek. Remote Exploitation of an Unaltered Passenger Vehicle, August 2015.

[16] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: a lightweight mac algorithm for microcontrollers. 2014.

[17] Irdin Pekaric, Clemens Sauerwein, Stefan Haselwanter, and Michael Felderer. A taxonomy of attack mechanisms in the automotive domain. *Computer Standards & Interfaces*, 78:103539, October 2021.

[18] Roger Antonio Pena, Mikel Pascual, Armando Astarloa, Daniel Uribe, and Jon Inchausti. Impact of MACsec security on TSN traffic. In *2022 37th Conference on Design of Circuits and Integrated Circuits (DCIS)*, pages 01–06, Pamplona, Spain, November 2022. IEEE.

[19] Mert D. Pesé, Jay W. Schauer, Junhui Li, and Kang G. Shin. S2-CAN: Sufficiently Secure Controller Area Network. In *Annual Computer Security Applications Conference*, pages 425–438, Virtual Event USA, December 2021. ACM.

[20] Dominik Püllen, Nikolaos A. Anagnostopoulos, Tolga Arul, and Stefan Katzenbeisser. Securing FlexRay-based in-vehicle networks. *Microprocessors and Microsystems*, 77:103144, September 2020.

[21] Gokay Saldamli, Levent Ertaul, and Asharani Shankaralingappa. Analysis of Lightweight Message Authentication Codes for IoT Environments. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 235–240, Rome, Italy, June 2019. IEEE.

[22] Alan Varghese. Automotive In-Vehicle Networks - Ethernet or SERDES? Well....It Is Complicated, January 2021.

[23] Giancarlo Vasta and Lucia Lo Bello. An innovative traffic management scheme for deterministic/event-based communications in automotive applications with a focus on Automated Driving Applications, October 2018.

[24] Dingwang Wang and Subramaniam Ganesan. Automotive Network Security. In *2021 IEEE International Conference on Electro Information Technology (EIT)*, pages 193–196, Mt. Pleasant, MI, USA, May 2021. IEEE.