Analyzing CAN's Timing under Periodically Authenticated Encryption

Mingqing Zhang[1], Philip Parsch[2], Henry Hoffmann[2] and Alejandro Masrur[1]
[1] *Department of Computer Science, TU Chemnitz, Germany*
[2] *Department of Computer Science, The University of Chicago, USA*

***Abstract*—With increasing connectivity, it has become easier to remotely access in-vehicle buses like CAN (Controller Area Network). This not only jeopardizes security, but it also exposes CAN's limitations. In particular, to reject replay and spoofing attacks, messages need to be authenticated, i.e., an authentication tag has to be included. As a result, messages become larger and need to be split in at least two frames due to CAN's restrictive payload. This increases the delay on the bus and, thus, some deadlines may start being missed compromising safety. In this paper, we propose a Periodically Authenticated Encryption (PAE) based on the observation that we do not need to send authentication tags with every single message on the bus, but only with a configurable frequency that allows meeting both safety and security requirements. Plausibility checks can then be used to detect whether non-authenticated messages sent in between two authenticated ones have been altered or are being replayed, e.g., the transmitted values exceed a given range or are not in accordance with previous ones. We extend CAN's known schedulability analysis to consider PAE and analyze its timing behavior based on an implementation on real hardware and on extensive simulations.**

***Index Terms*—automotive systems, CAN, AES, encryption, cybersecurity, authentication, safety, timing behavior, plausibility checks**

I. Introduction & Related Work

With increasing connectivity in different domains, cybersecurity is becoming a prevalent issue for researchers and practitioners in embedded systems [1] [2] [3]. In particular, in the automotive domain, vehicles increasingly exchange information with each other and with roadside/background infrastructure and, hence, it has become easier for attackers to gain access to in-vehicle buses like, particularly, the well-known CAN (Controller Area Network).

CAN was not conceived taking cybersecurity requirements into account and presents some serious limitations such as a restrictive payload (of only 8 bytes per frame) and a reduced bandwidth/transmission speed of typically 125 kbps or 500 kbps,[1] hindering the use of standard cybersecurity solutions.

Existing approaches to securing CAN either do not provide a sufficient level of security (e.g., they do not encrypt [4] [5] [6] [7] [8] or authenticate data [9] [10]) or require modifying the CAN protocol and/or controllers [4] [11] [9] [10] [12], which considerably increases costs and, hence, ends up jeopardizing competitiveness. This latter is also the case when considering more sophisticated buses like CAN-FD, FlexRay or Automotive Ethernet instead. As a result, there is a significant interest in alternative techniques to provide a sufficient level of security on CAN buses while still keeping costs low.

Since CAN follows a multi-master strategy, attackers can freely send messages upon gaining access to the bus. In particular, we concentrate three common attacks: Sniffing, Spoofing and Replay. Sniffing attacks can straightforwardly be avoided by encrypting data. To comply with CAN's payload of only 8 bytes per frame, one can use encryption techniques that do not increase the amount of data to be sent. However, this alone does not suffice to prevent replay and spoofing, where attackers capture and, most likely, alter legitimate (encrypted) messages with the aim of causing malfunction, even without being able to decrypt them. As a result, authentication needs to be added. On the other hand, authenticated messages need to be transmitted in two frames,[2] which leads to an increased delay on the bus compromising timing/safety.

II. Periodically Authenticated Encryption

In this paper, we propose an approach we denominate Periodically Authenticated Encryption (PAE) allowing for safety/security co-design on CAN buses. More specifically, we propose authenticating messages with a given configurable frequency, which allows us to reduce the associated overhead. Unauthenticated messages sent in between two authenticated ones are then validated by plausibility checks running on the different nodes. To illustrate this, let us consider the example of Electronic Stability Control (ESC). ESC aims to reduce an eventual loss of traction by selectively braking each individual wheel. To this end, ESC relies on sensors that periodically measure (e.g., every $10ms$) the rotational speeds of the wheels. If a wheel's rotational speed does not correlate with the vehicle's speed, this might be due to a loss in traction. The ECU then computes whether brakes need to be applied and by how much.

A spoofing attack targeted at the ESC, in which the measured rotational speeds are altered, can severely destabilize the vehicle. Note again that, even if we encrypt messages (from sensors to the ECU), the attacker can still capture a message and change random bits in the ciphertext (i.e., in the payload) replacing the CRC accordingly. The ECU will not be able to discern any alteration upon reception and will decrypt the message with altered data potentially leading to catastrophic consequences.

[1]CAN can also work at 1 Mbps, however, this is not possible within a vehicle due to high electromagnetic interference.

[2]The number of frames depends on the length of the authentication tag. The longer the tag, the more reliable the authentication, but also the more overhead is added. In this paper, we consider 8-byte authentication tags and, hence, authenticated messages require exactly two frames.

978-3-9819263-6-1/DATE22/©2022 EDAA

Authenticating messages prevents this kind of attacks, since the ECU can detect whether a message has been altered and discard it. However, authenticated messages require sending an authentication tag and, hence, they need to be sent within two frames (instead of only one) due to CAN's limited payload. This doubles the amount of data being sent on the bus and may potentially lead to deadline misses, i.e., sensor values do not reach the ECU in time, particularly since CAN's bandwidth is rather restrictive. As a consequence, the ESC may start malfunctioning compromising safety.

A. Plausibility checks

To maintain low costs and, at the same time, meet both security and safety requirements, we observe that the ECU can run a plausibility check to detect *anomalous* deviations in the rotational speed values reported by wheel sensors. That is, it will only accept values that are within a preconfigured range from a previously authenticated one. Messages containing values that exceed that range are automatically discarded (assuming that they have been altered by an external attacker). For example, assuming a passenger vehicle with a maximum linear acceleration of $4.4m/s^2$ and tires of $0.35m$ radius, we have that the rotational speed at a wheel can change at a rate of at most $4.4/0.35 \approx 12.57rad/s^2$ under normal traction conditions. At loss of traction the rotational speed might go up by around 10% to 20% depending on the vehicle. If values transmitted deviate from this, the corresponding messages can be discarded by the plausibility check. As a result, we do not need to authenticate every message sent by wheel sensors to the ECU, but only those that would otherwise not pass the plausibility check. Note that unauthenticated messages must still be encrypted to prevent sniffing attacks (provided that this does not increase the communication payload as discussed later).

B. Periodic authentication

In this paper, we consider that a CAN message m_i is authenticated periodically and introduce a parameter we call *authentication frequency* denoted by $\frac{1}{\alpha_i}$, where α_i indicates a given number of consecutive transmissions of m_i on the bus. More specifically, a frequency of 1 over 1 (short 1/1) implies that every message of m_i is authenticated. Similarly, a frequency of 1 over 10 (short 1/10) indicates that, within ten m_i messages sent, only one is authenticated, i.e., there are nine unauthenticated m_i messages between two authenticated ones. In the ESC example, with a period of $10ms$, this leads to a time interval of around $100ms$ (with variations due to the arbitration on the bus), in which messages are encrypted, but lack authentication.

The higher the authentication frequency, clearly, the higher the level of security attained (since more messages can be reliably verified). On the other hand, this also increases the overhead and, hence, affects timing on the bus. In addition, note that the authentication frequency also correlates with the level of safety. In particular, even though the plausibility check rejects altered messages, in the worst case, the ECU does not receive fully reliable updates on wheels' rotational speeds for some time interval (e.g., $100ms$ with an authentication frequency of 1/10). As a result, the authentication frequency should be chosen taking safety requirements into account yielding a safety/security co-design.

C. Used Encryption

We propose combining AES in CTR [13] and in GCM [14] mode to encrypt and/or authenticate data on CAN, respectively. Since data is encrypted/authenticated at each individual node before being sent, the CAN protocol does not need to be changed and, hence, we incur no additional costs.

AES-CTR is used to encrypt data without increasing its size, i.e., the ciphertext has the same number of bits as the plaintext. In our case, we consider that data is 8 bytes long, i.e., the full CAN payload is used.[3] AES-CTR can not only reject sniffing attacks, but it also makes replay and spoofing attacks more difficult. This is because a counter is used to encrypt data, which is increased (by a fixed amount at the corresponding sender and receivers) with every message sent. Since any previous transmission (stored and replayed by an attacker) uses a different counter value, this cannot be decrypted.

AES-GCM is based on AES-CTR and is used to generate 8-byte authentication tags that are sent with a given periodicity as discussed before. This allows for a higher level of security against replay and spoofing than AES-CTR alone. On the other hand, again, two data frames need to be transmitted with each authenticated message considerably increasing the overhead with respect to AES-CTR.

III. Schedulability Analysis

A. Without encryption/authentication

Let us first consider the case without encryption/authentication and denote by M the set of messages m_i sent over CAN with $1 \leq i \leq n$ and n being the total number of messages in M. Further, let us assume without loss of generality that all m_i are sorted in the order of decreasing priority, i.e., m_1 has higher priority than m_2, m_2 has higher priority than m_3 and so on.

We denote an m_i's transmission or communication time by c_i, which depends on CAN's bandwidth and the number of overhead, data and stuffing bits sent. Note that we consider the Intermission Field, i.e., the minimum separation between two consecutive frames on the bus, to be part of the overhead bits. In addition, we model the period of repetition of m_i by p_i and its deadline by d_i with $d_i \leq p_i$.

From [15] we know that M's schedulability is guaranteed, if the following holds for all i and k:

$$r_{i,k} \leq d_i + (k-1)p_i, \tag{1}$$

with $1 \leq k \leq 1 + \left\lfloor \frac{t_{busy} - d_i}{p_i} \right\rfloor$ and t_{busy} is the longest possible busy interval on CAN, i.e., the longest time interval without idling, given by the fixed point of:

$$t_{busy} = \sum_{i=1}^{n} \left\lceil \frac{t_{busy}}{p_i} \right\rceil c_i. \tag{2}$$

[3] If this is not the case, we assume that padding is used to enforce the full payload. This is because longer messages are generally more secure.

The variable $r_{i,k}$ in (1) represents the worst-case response time (WCRT), i.e., the maximum possible delay, by the k-th transmission of m_i in t_{busy}. In other words, m_i is schedulable on CAN, if it can meet its deadline each time it is sent within t_{busy}. Now, to compute $r_{i,k}$ for a given i and k, we proceed as follows:

$$r_{i,k} = b_i + k \cdot c_i + \sum_{j=1}^{i-1} \left\lceil \frac{r_{i,k}}{p_j} \right\rceil c_j, \tag{3}$$

which is again a fixed point computation. In (3), $b_i = \max_{i+1 \leq j \leq n}(c_j)$ is m_i's blocking time, i.e., the maximum delay that m_i may incur due to lower-priority messages.[4]

In (3), we assume that signals' propagation delays have been properly compensated at transceivers. In other words, if two or more nodes start sending simultaneously, the node with highest priority wins arbitration independent of its position/distance to other nodes on the bus.

B. With encryption/authentication

As discussed above, we use AES-CTR to encrypt messages without increasing the size of the ciphertext with respect to the plaintext. This way, any encrypted m_i can be sent within one frame. On the other hand, to enforce using CAN's full payload, we add padding to the plaintext, if this is shorter than 8 bytes. The transmission/communication time of any encrypted m_i is hence equal to c_{max}, assuming that a maximum possible number of stuffing bits are sent.

As discussed above, we assume that every m_i in M is authenticated with a frequency of $1/\alpha_i$. That is, one over α_i messages of type m_i will be authenticated, where $\alpha_i \geq 1$ is an integer number expressing a given number of consecutive m_i messages sent. To model the workload in this case, we note that an additional frame is sent every α_i single-frame messages of m_i. That is, a single-frame message with a period of $\alpha_i \cdot p_i$ is sent. As a result, we have that the busy interval as follows:

$$t'_{busy} = \sum_{i=1}^{n} \left\lceil \frac{t'_{busy}}{p_i} \right\rceil c_{max} + \sum_{i=1}^{n} \left\lceil \frac{t'_{busy}}{\alpha_i \cdot p_i} \right\rceil c_{max}. \tag{4}$$

In other words, since CAN's utilization increases when authenticating messages, the busy interval also increases from t_{busy} to t'_{busy}. In addition, the WCRT of the k-th message of type m_i is computed as follows:

$$\begin{aligned} r'_{i,k} &= \hat{b}_i + k \cdot c_{max} + \left\lceil \frac{k}{\alpha_i} \right\rceil c_{max} + \sum_{j=1}^{i-1} \left\lceil \frac{r'_{i,k}}{p_j} \right\rceil c_{max} \\ &\quad + \sum_{j=1}^{i-1} \left\lceil \frac{r'_{i,k}}{\alpha_j \cdot p_j} \right\rceil c_{max}. \end{aligned} \tag{5}$$

Clearly, the above expression derives from (3). $\lceil \frac{k}{\alpha_i} \rceil c_{max}$ considers the additional overhead by authenticated m_i messages, whereas $\lceil \frac{r'_{i,k}}{\alpha_j \cdot p_j} \rceil c_{max}$ accounts for the authentication overhead by higher-priority m_j messages with $1 \leq j \leq i-1$.

[4]Since ongoing transmission cannot be interrupted, CAN follows a non-preemptive scheduling.

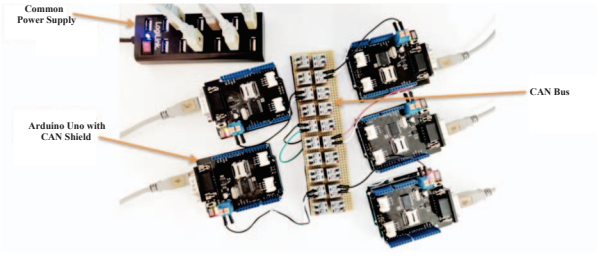


Fig. 1: Implementation on Arduino UNO/CAN-BUS Shields

TABLE I: Timings in our implementation

Symbol	Description	Value [μs]
t_{ECU}	Overhead on node accounts for CAN stack	125
t_{CAN}	Average transmission time of frames sent	250
t_{GCM}	Time for encryption/decryption by AES-GCM	878
t_{CTR}	Time for encryption/decryption by AES-CTR	533

Further, even if a lower-priority m_j with $i+1 \leq j \leq n$ may also be authenticated and, hence, also consist of two frames, m_i can only be blocked by the first m_j's frame, i.e., $\hat{b}_i$ is equal to c_{max}, i.e., transmission time of one encrypted frame.

The schedulability of authenticated messages with frequency of $1/\alpha_i$ can be guaranteed, if the following holds for $1 \leq i \leq n$ and $1 \leq k \leq 1 + \left\lfloor \frac{t'_{busy} - d_i}{p_i} \right\rfloor$ with t'_{busy} given as per (4):

$$r'_{i,k} \leq d_i + (k-1) \cdot p_i.$$

Finally, if every single message is authenticated, i.e., $\alpha_i = 1$ for all i, it is easy to see that (4) and (5) turn to (6) and (7):

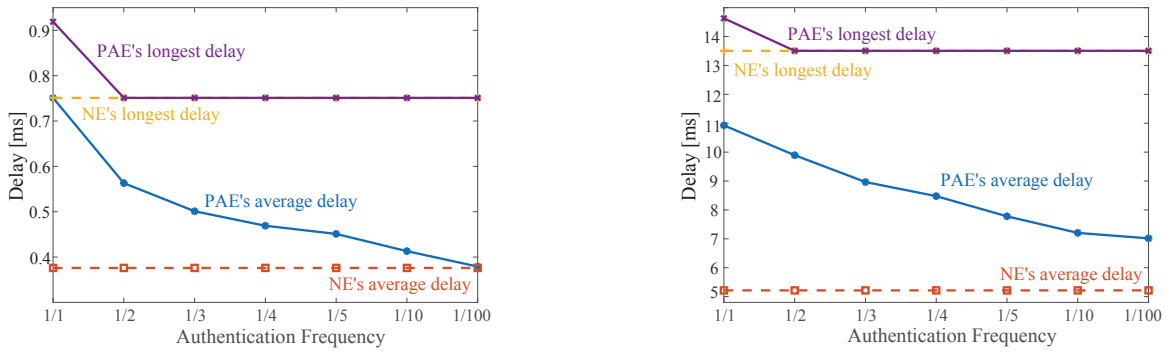
$$t''_{busy} = \sum_{i=1}^{n} \left\lceil \frac{t''_{busy}}{p_i} \right\rceil 2c_{max}, \tag{6}$$

$$r''_{i,k} = 2c_{max} \left(1 + k + \sum_{j=1}^{i-1} \left\lceil \frac{r''_{i,k}}{p_j} \right\rceil \right). \tag{7}$$

IV. Implementation and Evaluation

We implemented the proposed approach on Arduino UNO boards equipped with CAN-BUS Shields, see Fig. 1. Each node transmits around 10,000 CAN messages to collect relevant timing information. The resulting average values are presented in Table I. Our approach has an average end-to-end delay of $533ms$ and $878ms$ when applying AES-CTR and AES-GCM, respectively. Note that t_{ECU}, t_{CTR} and t_{GCM} are rather constant times that depend on the platform used (in our case Arduino UNO/CAN-BUS Shield), whereas t_{CAN} depends on CAN's bandwidth and contention on the bus. In the next section, we evaluate the proposed approach in terms of improving timing under different levels of contention.

In order to evaluate our PAE approach with respect to the case of no encryption (NE), a CAN simulation in OMNeT++ [17] [18] is used. In particular, we simulated BMW E90's CAN messages from [16], for which we used t_{ECU}, t_{CTR} and t_{GCM} obtained in Section IV.



(a) The highest-priority message (i.e., ID 0x0A8)

(b) The lowest-priority message (i.e., ID 0x581)

Fig. 2: Delay under different authentication frequencies for BMW E90's message set [16]

Fig. 2a and Fig. 2b show the timing behaviors of the highest-priority and lowest-priority messages. As expected, the average delay decreases rapidly as the authentication frequency decreases from 1/1 to 1/100, i.e., from the case where every message is authenticated to the case where only one over 100 messages is authenticated. It can be noticed that the highest-priority message reaches the minimum possible average delay under PAE at an authentication frequency of 1/100, however, this is not the case for the lowest-priority message.

Further, as also shown in Fig. 2a and Fig. 2b, PAE's longest delay for both the highest and lowest priority already minimizes at an authentication frequency of 1/2, i.e., it suffices to authenticate every second message to reach the same timing behavior as in the NE case. This is due to the relatively low utilization of this message set. Clearly, for a higher utilization, it will be necessary to further decrease the authentication frequency to achieve similar results.

V. Conclusion

In this paper, we proposed our Periodically Authenticated Encryption (PAE) approach to protect CAN buses against replay, spoofing and sniffing attacks and, at the same time, meet timing requirements. In contrast to approaches from the literature, the proposed technique does not require modifying CAN and, hence, it does not increase costs.

The idea is that plausibility checks can detect altered messages by a replay/spoofing attack, e.g., transmitted values exceed a preconfigured range. As a result, not every message needs to be authenticated, but only those that would otherwise not pass plausibility checks introducing an *authentication frequency*. This allows reducing the overhead incurred by authentication and, hence, allows us to preserve CAN's timing behavior and guarantee safety.

We extended CAN's known schedulability analysis to this case and illustrated PAE's advantages based on an implementation on real hardware and by means of OMNeT++ simulation of a realistic message set. Overall, for this message set, we showed that authenticating every second message on CAN already suffices to achieve the same timing behavior as in the case of no encryption (NE).

References

[1] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in Embedded Systems: Design Challenges," *ACM Transactions On Embedded Computing Systems*, 2004.

[2] D. Serpanos and A. Voyiatzis, "Security Challenges in Embedded Systems," *ACM Transactions On Embedded Computing Systems*, 2013.

[3] T. Wollinger, J. Guajardo, and C. Paar, "Security on FPGAs: State-of-the-Art Implementations and Attacks," *ACM Transactions On Embedded Computing Systems*, 2004.

[4] A. Herrewege, D. Singelée, and I. Verbauwhede, "CanAuth - A Simple Backward Compatible Broadcast Authentication Protocol for CAN Bus," in *Embedded Security in Cars Conference*, 2011.

[5] S. Fassak, Y. Idrissi, N. Zahid, and M. Jedra, "A secure protocol for session keys establishment between ecus in the can bus," in *International Conference on Wireless Networks and Mobile Communications*, 2017.

[6] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security Authentication System for In-vehicle Network," *SEI Technical Review*, 2015.

[7] Z. King and S. Yu, "Investigating and Securing Communications in the Controller Area Network (CAN)," in *2017 International Conference on Computing, Networking and Communications*, 2017.

[8] J. Yeom and S. Seo, "A Methodology of CAN Communication Encryption Using a shuffling algorithm," in *International Conference on Connected and Autonomous Driving (MetroCAD)*, 2020.

[9] W. Farag, "CANTrack: Enhancing Automotive CAN Bus Security using Intuitive Encryption Algorithms," in *International Conference on Modeling, Simulation, and Applied Optimization*, 2017.

[10] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, and K. Oishi, "A Method of Preventing Unauthorized Data Transmission in Controller Area Network," in *IEEE 75th Vehicular Technology Conference*, 2012.

[11] B. Groza, P.-S. Murvay, A. Herrewege, and I. Verbauwhede, "LiBrA-CAN: a Lightweight Broadcast Authentication protocol for Controller Area Networks," in *International Conference on Cryptology and Network Security*, 2012.

[12] O. Hartkopp, R. Cornel, and R. Schilling, "MaCAN - Message authenticated CAN," in *Escar Conference on Embedded Security in Cars*, 2012.

[13] M. J. Robshaw, "Stream ciphers technical report tr-701," 2009.

[14] J. Daemen and V. Rijmen, *The Design of Rijndael*, 2002.

[15] R. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised," *Real-Time Systems*, 2007.

[16] R. Buttigieg, M. Farrugia, and C. Meli, "Security Issues in Controller Area Networks in Automobiles," in *International Conference on Sciences and Techniques of Automatic Control and Computer Engineering*, 2017.

[17] J. Matsumura, Y. Matsubara, H. Takada, M. Oi, M. Toyoshima, and A. Iwai, "A Simulation Environment based on OMNeT++ for Automotive CAN–Ethernet Networks," in *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems*, 2013.

[18] K. Kawahara, Y. Matsubara, and H. Takada, "A Simulation Environment and Preliminary Evaluation for Automotive CAN-Ethernet AVB Networks," *CoRR*, 2014.