Work in Progress: Schedulability Analysis of CAN and CAN FD Authentication

Omolade Ikumapayi*, Habeeb Olufowobi†, Jeremy Daily‡, Tingting Hu§, Ivan Cibrario Bertolotti¶, Gedare Bloom*
*University of Colorado Colorado Springs, †University of Texas at Arlington, ‡Colorado State University, Fort Collins,
§University of Luxembourg, ¶National Research Council of Italy
{oikumapa, gbloom}@uccs.edu

***Abstract*—Ensuring the data integrity of messages transmitted over the Controller Area Network (CAN) bus and other vehicular networks is achieved through the implementation of cryptographic authentication protocols. However, these protocols raise concerns about a significant increase in response time due to the restrictions on CAN frame size and bandwidth. This paper presents a comprehensive analysis of the impact on response time of CAN and CAN Flexible Data-rate (CAN FD) messages with the implementation of cryptographic message authentication codes (MACs) and the periodic transmission of these codes. Our evaluation is based on a randomized schedulability experiment to provide insights into the overhead incurred by adding authentication to the frame payloads.**

***Index Terms*—Controller Area Network, CAN FD, Response Time Analysis**

I. Introduction

The AUTOSAR specification for secure onboard communication (SecOC) standardizes the conditions for authenticating protocol data units, including the specific length of the cryptographic message authentication code (MAC) [1]. The authentication comprises a Freshness Value (FV) and an Authenticator. The integration of MAC leads to an increase in transmission time especially when extra data frames are required to transmit the excess messages that cannot be accommodated in the 8-byte data payload. The overhead from extra frames significantly increases the transmission time imposed by authentication. Due to the significant overhead, an alternative is to transmit periodic MACs that combines selective authentication skipping for some of the messages and grouping the authentication bits as a batch [2], [3]. The introduction of CAN FD aims to address the limitations of the original CAN protocol by providing increased payload length and bandwidth. However, it is still essential to consider the overhead introduced by adding the MAC. In this work, we present a thorough approach to bound the impact of authentication on CAN with response time analysis (RTA) for CAN schedulability. The contributions of this paper are:

- RTA formulation for CAN FD;
- RTA formulation for adding MAC and periodic MAC to CAN and CAN FD;
- and evaluation of SecOC authentication for CAN and CAN FD with randomized schedulability experiments.

This work is supported by NSF CNS-2046705 and Colorado Bill SB18-086.

II. Related Work

The impact of authentication protocols on the timely arrival of messages for automotive applications is a significant concern that is well-studied. Nürnberger et al. [4] proposed a backward-compatible authentication mechanism called vatiCAN, which utilizes a lightweight keyed-Hash Message Authentication Code (HMAC). They evaluated the authentication overhead and concluded that their implementation, which authenticates each CAN frame separately, would impose an impractical bandwidth overhead. Previous studies have computed the worst-case transmission time of CAN FD [5], [6]. However, formulations are flawed due to using an outdated pre-specification prior to ISO standardization which has been updated [5], and use of the obsolete CAN transmission time [6] that has been revised [7]. In addition, these earlier studies did not consider the extended format (29-bit identifier). In this work, we provide an analysis of the transmission time of CAN FD for the revised version of the base format and the formulation of the extended format. We adjusted the transmission time applicable to both CAN and CAN FD to account for the addition of MACs as well.

III. Classical CAN Response Time Analysis

We adopt the widely used response time analysis for CAN devised by Tindell et al. [8] and revised by Davis et al. [7]. The traditional CAN base format transmission time of a message M_i with payload length of D_i bytes is given by:

$$C_{D_i}^{(C,b)} = (55 + 10D_i)\tau_{bit}, \tag{1}$$

τ_{bit} is the time needed to transmit a single bit. For the extended format, the worst-case transmission time is given by:

$$C_{D_i}^{(C,e)} = (80 + 10D_i)\tau_{bit} \tag{2}$$

where the superscript (C, b) and (C, e) denote the classical base and extended frame format respectively. A message instance q in the busy interval has a WCRT of $R_i(q)$. These variables are found by solving

$$R_i(q) = J_i + w_i(q) - qP_i + C_{D_i} \tag{3}$$

$$Q_i = \left\lceil \frac{t_i + J_i}{P_i} \right\rceil \tag{4}$$

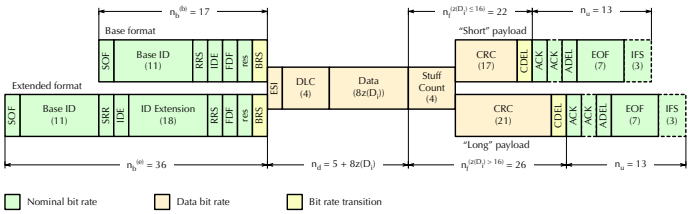

Fig. 1: CAN FD Frame Format (Stuff Bits Not Shown)

J_i is queueing jitter of the frame, P_i is the period and Q_i is the number of instances for M_i that release during the busy interval. The worst-case delay w_i, given an error model is found by solving the recurrence

$$w_i^{n+1}(q) = B_i + E(w_i^n + C_{D_i}) + qC_{D_i} + I_i \tag{5}$$

starting with $w_i^0(q) = B_i + qC_{D_i}$ and terminating at $w_i^{n+1}(q) = w_i^n(q)$. The blocking B_i caused by lower-priority messages is:

$$B_i = \max_{k>i}(C_{D_k}), \tag{6}$$

and I_i for *interference* caused by higher-priority messages that beat q during arbitration:

$$I_i = \sum_{k<i} \left\lceil \frac{w_i + J_k + \tau_{bit}}{P_k} \right\rceil C_{D_k}. \tag{7}$$

The length of the level-i busy interval is given by t_i, and it is found by solving the recurrence relation

$$t_i^{n+1} = B_i + E_i(t_i^n) + \sum_{k\leq i} \left\lceil \frac{t_i^n + J_k}{P_k} \right\rceil C_{D_k} \tag{8}$$

starting with $t_i^0 = C_{D_i}$ and terminating at $t_i^{n+1} = t_i^n$. With 31 bits for the error signal, message error handling, E_i is formulated as

$$E_i(t_i) = \Big(31\tau_{bit} + \max_{k\geq i}(C_{D_k})\Big)F(t_i) \tag{9}$$

IV. CAN FD Response Time Analysis

The CAN FD frames accommodate up to 64 data bytes. The payload length (DLC), however is still encoded as a 4-bit value, so only a few predefined lengths are available. For this reason, we define a function $z(D_i)$ that determines the smallest feasible payload length for a frame's data payload to accommodate D_i bytes. The function returns a value from the set $\{0, 1, \ldots, 8, 16, 20, 24, 32, 48, 64\}$ depending on M_i's payload length. In addition, CAN FD optionally supports a bitrate switch (BRS) to a higher bitrate for part of the frame, where the bit transmission time becomes $\tau_{dbit} \leq \tau_{bit}$. As shown in Figure 1, a CAN FD frame can be divided into four parts for worst-case C_{D_i} calculation:

1) n_b bits between the SOF and BRS bit, transmitted with on-demand bit stuffing at the nominal bitrate.
2) n_d bits transmitted with on-demand bit stuffing at the data bitrate after BRS and before the CRC field.
3) n_f bits in the CRC field, transmitted with a fixed bit stuffing at the data bitrate, except for the CDEL bit (the last bit of the CRC field)
4) n_u bits after the CRC field without bit stuffing at the nominal bitrate. In CAN FD this part may consist of two ACK slots, instead of one like in classical CAN according to the standard (ISO 11898-1 §10.4.2.7).

$n = n_b + n_d + n_f + n_u$ is the total number of unstuffed bits in the frame. The worst-case number of stuff bits s_b needed by the first n_b bits can be calculated considering the worst-case pattern for on-demand bit stuffing. We obtain:

$$s_b = \left\lfloor \frac{n_b - 1 - 1}{4} \right\rfloor. \tag{10}$$

The two corrective terms -1 in the numerator of (10) are needed to take into account that the primer of the worst-case bit stuffing pattern consists of 5 bits instead of 4, and to avoid counting in s_b a stuff bit to be injected immediately after BRS, because that stuff bit is transmitted at the data rather than the nominal bitrate.

The number of unstuffed bits, including BRS after the last stuff bit transmitted at the nominal bitrate is given by:

$$r_b = (n_b - 1) - 4s_b. \tag{11}$$

A value $r_b = 4$ indicates that a stuff bit must be transmitted, at the data bitrate, after BRS. The worst-case number of stuff bits s_d needed by the next n_d bits is then given by:

$$s_d = \left\lfloor \frac{r_b + n_d - 1}{4} \right\rfloor. \tag{12}$$

In (12) the -1 at the numerator avoids counting an on-demand stuff bit to be injected immediately before the CRC field.

Finally, the number of fixed stuff bits s_f needed by the n_f bits in the CRC field is:

$$s_f = \left\lceil \frac{n_f}{4} \right\rceil. \tag{13}$$

In this equation, we use a ceiling instead of a floor operator to count the stuff bit injected before the first bit of the CRC field. Subsequent n_u bits up to the end of the frame are transmitted without bit stuffing, so it (trivially) is $s_u = 0$.

The total worst-case transmission time C_{D_i} of a message M_i can be expressed as the sum of the contributions of the four parts of the frame just described, that is

$$C_{D_i}^{(F)} = (n_b + s_b + n_u)\tau_{bit} + (n_d + s_d + n_f + s_f)\tau_{dbit} \tag{14}$$

where the superscript (F) is used to denote CAN FD. Substituting the values from Eq. (10–13) into (14) four cases are possible, depending on the frame format and payload length D_i, in bytes:

$$\begin{aligned} C_{D_i}^{(F,b,D_i\leq 16)} &= 33\tau_{bit} + (35 + 10z(D_i))\tau_{dbit} \\ C_{D_i}^{(F,b,D_i>16)} &= 33\tau_{bit} + (40 + 10z(D_i))\tau_{dbit} \\ C_{D_i}^{(F,e,D_i\leq 16)} &= 57\tau_{bit} + (34 + 10z(D_i))\tau_{dbit} \\ C_{D_i}^{(F,e,D_i>16)} &= 57\tau_{bit} + (39 + 10z(D_i))\tau_{dbit} \end{aligned} \tag{15}$$

Here the superscript indicates the frame format (F, b for base and F, e for extended) and the size of the data payload below (inclusive) or above 16 bytes. The remainder of the RTA for CAN FD is unmodified.

V. MAC Response Time Analysis

The inclusion of MAC requires the consideration of the extra frame generated when the data payload exceeds the maximum frame's payload length D_{max}. We introduce β_i as the number of full message frames required for transmitting the authenticator and FV with the original frame's payload and α_i as the number (at most one) of partial message frames. The adjusted worst-case transmission time is given by:

$$\widehat{C_{D_i}} = \beta_i C_{D_{\max}} + \alpha_i C_{(D_i+A_i+F_i) \mod D_{\max}}. \tag{16}$$

where β_i and α_i are formulated as:

$$\beta_i = \left\lfloor \frac{D_i + A_i + F_i}{D_{\max}} \right\rfloor, \tag{17}$$

$$\alpha_i = \left\lceil \frac{D_i + A_i + F_i}{D_{\max}} \right\rceil - \beta_i. \tag{18}$$

We adjusted the blocking and inference to account for the addition of MAC due to the addition of authentication data in the same frame Hence, the new blocking is given as:

$$\widehat{B_i} = \max_{k>i}(\max(\lceil \beta_k/(\beta_k+1) \rceil C_{D_{\max}}, C_{(D_k+A_k+F_k) \mod D_{\max}})). \tag{19}$$

The additional message frames then modify Eq. 4 to give $\widehat{Q_i}$ by multiplying Q_i by $(\beta_i + \alpha_i)$. Due to the extra instances generated, the level-i busy interval from Eq. 8 changes to $\widehat{t_i}$, and it is found by solving the recurrence relation

$$\widehat{t_i^{n+1}} = \widehat{B_i} + E_i(\widehat{t_i^n}) + \sum_{k \le i} \left\lceil \frac{\widehat{t_i^n} + J_k}{P_k} \right\rceil \widehat{C_{D_k}} \tag{20}$$

Another task is to modify the WCRT $R_i(q)$ of message instance q (Eq. 3). By adjusting the waiting time $w_i(q)$ from the recurrence in Eq. 5 to include the interference due to the extra message instances, the new recurrence becomes

$$\begin{aligned} \widehat{w_i^{n+1}}(q) = \widehat{B_i} &+ E(\widehat{w_i^n} + \widehat{C_{D_i}}) \\ &+ (q \mod (\beta_i + \alpha_i)) C_{D_{\max}} \\ &+ \left\lfloor \frac{q}{\beta_i + \alpha_i} \right\rfloor \widehat{C_{D_i}} + \widehat{I_i} \end{aligned} \tag{21}$$

where

$$\widehat{I_i} = \sum_{k<i} \left\lceil \frac{\widehat{w_i} + J_k + \tau_{bit}}{P_k} \right\rceil \widehat{C_{D_k}}. \tag{22}$$

The final WCRT by modifying Eq. 3 is given by:

$$\begin{aligned} \widehat{R_i}(q) = J_i + \widehat{w_i(q)} &- \left\lfloor \frac{q}{\beta_i + \alpha_i} \right\rfloor P_i \\ &+ \left(1 - \left\lfloor \frac{(q \mod (\beta_i + \alpha_i)) + 1}{\beta_i + \alpha_i} \right\rfloor \right) C_{D_{\max}} \\ &+ \left\lfloor \frac{(q \mod (\beta_i + \alpha_i)) + 1}{\beta_i + \alpha_i} \right\rfloor \Gamma_i \end{aligned} \tag{23}$$

where

$$\Gamma_i = \alpha_i (C_{(D_i+A_i+F_i) \mod D_{\max}}) + ((1-\alpha_i)(\beta_i - 1) C_{D_{\max}}). \tag{24}$$

We introduce Γ_i with Eq. 24 yielding either the transmission time of a partial message in case $\alpha_i = 1$ or yielding $C_{D_{\max}}$ in case $\alpha_i = 0$ and $\beta_i > 1$.

VI. Periodic MAC Response Time Analysis

To accommodate periodic MACs, additional message frames are generated that will carry the MAC at a periodic rate ρ_i. Also, when message frames are sporadic, the MAC is sent every ρ_i / P_i transmissions of message M_i. We assume that the periodic MAC is sent independently (though with the same ID) from data payloads. The adjusted transmission time for period MAC is given by:

$$\widetilde{C_{D_i}} = \widetilde{\beta}_i C_{D_{\max}} + \widetilde{\alpha}_i C_{(A_i+F_i) \mod D_{\max}}. \tag{25}$$

and the blocking time is

$$\begin{aligned} \widetilde{B_i} = \max_{k>i}(\max(C_{D_k}, &\lceil \beta_k/(\beta_k+1) \rceil C_{D_{\max}}, \\ &C_{(A_k+F_k) \mod D_{\max}})). \end{aligned} \tag{26}$$

Hence, we first modify Eq. 17 and Eq. 18 to determine the number of full frames and (at most one) partial frames needed for authentication as

$$\widetilde{\beta}_i = \left\lfloor \frac{A_i + F_i}{D_{\max}} \right\rfloor \tag{27}$$

$$\widetilde{\alpha}_i = \left\lceil \frac{A_i + F_i}{D_{\max}} \right\rceil - \widetilde{\beta}_i \tag{28}$$

the modification of Q_i from Eq. 4 ($\widehat{Q_i}$ respectively) becomes

$$\widetilde{Q_i} = \left\lceil \frac{\widetilde{t_i} + J_i}{P_i} \right\rceil + \left\lceil \frac{\widetilde{t_i} + J_i}{\rho_i} \right\rceil (\widetilde{\beta}_i + \widetilde{\alpha}_i). \tag{29}$$

The level-i busy interval is now $\widetilde{t_i}$ found by solving the recurrence relation

$$\begin{aligned} \widetilde{t_i^{n+1}} = \widetilde{B_i} &+ E_i(\widetilde{t_i^n}) \\ &+ \sum_{k \le i} \left(\left\lceil \frac{\widetilde{t_i^n} + J_k}{P_k} \right\rceil C_{D_k} + \left\lceil \frac{\widetilde{t_i^n} + J_k}{\rho_k} \right\rceil \widetilde{C_{D_k}} \right) \end{aligned} \tag{30}$$

starting with $\widetilde{t_i^0} = C_{D_i} + \widetilde{C_{D_i}}$ and terminating at $\widetilde{t_i^{n+1}} = \widetilde{t_i^n}$.

We again need to adjust the waiting time $\widehat{w_i}(q)$ from the recurrence in Eq. 21 to include the interference due to the periodic generation of extra instances, hence

$$\widetilde{w_i^{n+1}}(q) = \widetilde{B_i} + E(\widetilde{w_i^n} + C_{D_i}) + \left\lfloor \frac{q}{\rho_i/P_i + (\widetilde{\beta_i} + \widetilde{\alpha_i})} \right\rfloor \left(\frac{\rho_i}{P_i} C_{D_i} + \widetilde{C_{D_i}} \right) + \min(q', \tfrac{\rho_i}{P_i}) C_{D_i} + \max(0, q' - \tfrac{\rho_i}{P_i}) C_{D_{max}} + \widetilde{I_i} \quad (31)$$

The recurrence is solved starting with $\widetilde{w_i^0}(q) = \widetilde{B_i} + qC_{D_i}$ and terminating at $\widetilde{w_i^{n+1}}(q) = \widetilde{w_i^n}(q)$.

$$q' = q \mod (\rho_i/P_i + (\widetilde{\beta_i} + \widetilde{\alpha_i})) \quad (32)$$

$$\widetilde{I_i} = \sum_{k<i} \left(\left\lceil \frac{\widetilde{w_i} + J_k + \tau_{bit}}{P_k} \right\rceil C_{D_k} + \left\lceil \frac{\widetilde{w_i} + J_k + \tau_{bit}}{\rho_k} \right\rceil \widetilde{C_{D_k}} \right). \quad (33)$$

The periodic MAC WCRT is given by:

$$\widetilde{R_i}(q) = J_i + \widetilde{w_i}(q) - \left\lfloor \frac{q}{\rho_i/P_i + \widetilde{\beta_i} + \widetilde{\alpha_i}} \right\rfloor \rho_i + \max(C_{D_i}, \widetilde{C_{D_i}}). \quad (34)$$

VII. Evaluation

Our evaluation assesses the impact of message authentication on real-time performance using synthetic workloads. We generated 1000 different sets of messages with ranges of the period. The DLC values range from 1 to 8 bytes for CAN and up to 64 bytes for CAN FD. The period for each message is selected from a set of predefined values 5, 10, 100, 1000, and 5000ms. Also, we varied the bus utilization level from 10 to 90%. Message IDs (priorities) are assigned rates monotonically with implicit deadlines. We applied the CAN RTA to each message set created to establish a baseline and proceeded to compute the MAC and periodic MAC RTA using the same sets of messages. For both approaches, we included an authenticator utilizing SecOC Profile 1, specifically the 24Bit-CMAC-8Bit-FV. If the resulting payload exceeds the 8 bytes and 64 bytes $D_{\max}$ for CAN and CAN FD respectively, then we generate an additional frame, i.e., $\beta_i + \alpha_i = 2$.

For the periodic MAC approach, we investigated the impact of adding an authenticator to each message at different periodic rates: $\rho_i = 2P_i$ and $\rho_i = 10P_i$. We also consider $\rho_i = P_i$, where an authenticator frame is generated for each message instance. The results are presented in Fig. 2, which shows the percentage of message sets that can be scheduled under different authentication schemes when the CAN bitrate is set to 250kbps. As anticipated, the MAC and periodic MAC with $\rho_i = P_i$ demonstrate the worst performance. The periodic MAC approaches that skip more messages in the authentication scheme show favorable results until bus utilization reaches 70%. At 80%, the $\rho_i = 2P_i$ scheme fails to converge, while at 90% utilization, similar behavior is observed with $\rho_i = 10P_i$. For CAN FD, Fig. 3 displays the percentage of message sets schedulable with a bus speed of 250kbps and 4Mbps for bitrate switching (for τ_{bit} and τ_{dbit}) using Eq. (15) to generate their transmission times. We observed a decline in schedulability when bus utilization reached 65% for the periodic MAC and $\rho_i = P_i$. Additionally, the MAC and $\rho_i = 2P_i$ method failed to converge at 90% bus utilization. The periodic MAC does perform the best with sufficiently high enough period (e.g., $\rho_i = 10 * P_i$) at high bus utilization.

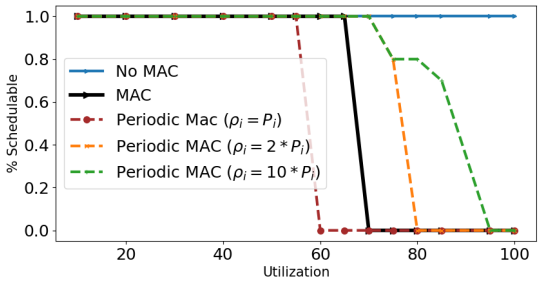

Fig. 2: Schedulability as a percent of messages meeting deadlines ($R_i < P_i$) of random message sets for CAN with varying bus utilization and authentication schemes.

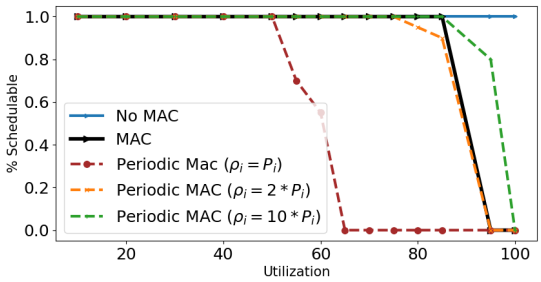

Fig. 3: Schedulability of random message sets for CAN FD with varying bus utilization and authentication schemes at $\tau_{bit} = 0.002$, $\tau_{dbit} = 0.00025$.

VIII. Conclusion

This paper proposes a novel and comprehensive approach to assessing the effect of authentication schemes on the real-time performance of messages transmitted over CAN and CAN FD, based on response time analysis. Future work can extend this approach to consider other schemes for authentication.

References

[1] C. AUTOSAR, "Specification of secure onboard communication," *AUTOSAR CP Release*, vol. 4, no. 1, 2017.

[2] M. Zhang, P. Parsch, H. Hoffmann, and A. Masrur, "Analyzing can's timing under periodically authenticated encryption," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022.

[3] J. Daily, D. Nnaji, and B. Ettlinger, "Securing CAN Traffic on J1939 Networks," in *Workshop on Automotive and Autonomous Vehicle Security*. Internet Society, Feb. 2021.

[4] S. Nürnberger and C. Rossow, "–vatican–vetted, authenticated can bus," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 106–124.

[5] U. D. Bordoloi and S. Samii, "The frame packing problem for can-fd," in *2014 IEEE Real-Time Systems Symposium*. IEEE, 2014, pp. 284–293.

[6] R. De Andrade, K. N. Hodel, J. F. Justo, A. M. Laganá, M. M. Santos, and Z. Gu, "Analytical and experimental performance evaluations of can-fd bus," *IEEE Access*, vol. 6, pp. 21 287–21 295, 2018.

[7] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[8] K. Tindell, H. Hanssmon, and A. J. Wellings, "Analysing real-time communications: Controller area network (CAN)." in *RTSS*, 1994.