Network Scheduling for Secure Cyber-Physical Systems

Vuk Lesi, Ilija Jovanov and Miroslav Pajic
Department of Electrical & Computer Engineering
Duke University
{vuk.lesi, ilija.jovanov, miroslav.pajic}@duke.edu

***Abstract*—Existing design techniques for providing security guarantees against network-based attacks in cyber-physical systems (CPS) are based on continuous use of standard cryptographic tools to ensure data integrity. This creates an apparent conflict with common resource limitations in these systems, given that, for instance, lengthy message authentication codes (MAC) introduce significant overheads. We present a framework to ensure both timing guarantees for real-time network messages and Quality-of-Control (QoC) in the presence of network-based attacks. We exploit physical properties of controlled systems to relax constant integrity enforcement requirements, and show how the problem of feasibility testing of intermittently authenticated real-time messages can be cast as a mixed integer linear programming problem. Besides scheduling a set of real-time messages with predefined authentication rates obtained from QoC requirements, we show how to optimally increase the overall system QoC while ensuring that all real-time messages are schedulable. Finally, we introduce an efficient runtime bandwidth allocation method, based on opportunistic scheduling, in order to improve QoC. We evaluate our framework on a standard benchmark designed for CAN bus, and show how an infeasible message set with strong security guarantees can be scheduled if dynamics of controlled systems are taken into account along with real-time requirements.**

I. Introduction

Recent years have witnessed significant increase in the number of security related incidents in cyber-physical systems (CPS). For instance, automotive attacks (e.g., [1], [2]) as well as the capturing of the RQ-170 Sentinel US drone [3] have illustrated that even safety-critical automotive and military CPS can be tampered with or completely hijacked. One of the main reasons for such dire situations is the expansion in network connectivity and complete reliance on perimeter security in these systems. Thus, by compromising an internal system component and utilizing interconnections between the components, an attacker could easily launch attacks over low-level networks used for real-time communication of safety-critical and control-related packets. This, in turn, could allow him to force the controlled physical process into any desired state as illustrated in [1], [2] for automotive systems.

Some of these network-based attacks, such as the *Man-in-the-Middle* (MitM) attacks, can be avoided with the use of standard cryptographic tools. For example, in CAN networks, a common approach is to add a message authentication code (MAC) to *all* transmitted measurements, in order to authenticate data and ensure integrity of received packages. On the other hand, CPS are often resource constrained, and might not be able to handle the continuous overhead caused by computation and communication of such codes for a sufficient number of sensors. For example, as presented in [4], [5], adding more than 30 MAC bits to CPS systems based on CAN networks may not be feasible due to the message length limitation (e.g., only 64 payload bits in the basic CAN protocol); yet, splitting them into several communication packets can significantly increase the message transmission time and reduce system/control performance.

This conflicting set of requirements, between the overhead introduced with the use of security mechanisms and the obtained security guarantees, is common for security-related research. However, to the best of our knowledge, no work provides direct relationship between the use of system resources and the overall system performance, in terms of its main functionality, in the presence of attacks. For example, [6] explores opportunistic execution of security-related tasks on top of existing legacy systems in order to integrate security mechanisms. The authors formulate optimization problems around adaption of parameters of security-related tasks, while maintaining schedulability of existing non-security-related tasks. In [7] a novel scheduling algorithm is proposed to jointly take into account security and real-time requirements for embedded systems. The approach is evaluated only abstractly, by the measure of an abstract *security level* with no direct relationship with system performance in the presence of attacks. In [8], a similarly defined security level is maximized by optimally choosing active security services with respect to schedulability conditions.

In this work, we focus on providing security guarantees, in terms of Quality-of-Control (QoC), for control components in CPS in the presence of network-based attacks. We assume that the attacker may have access to the low-level network and could inject false sensor measurements and actuator commands. While such attacks on actuator commands cannot stay undetected (i.e., stealthy), by changing messages from a subset of sensors a stealthy attacker can force the controlled plant far from the desired operating point through the actions of the controller [9], [10]. On the other hand, we have recently shown that it is not necessary to continuously ensure data integrity for satisfiable control performance in the presence of attacks [11], [12]. Since we exploit limitations imposed on the attacker by the physical laws governing behavior of

This work was supported in part by the NSF CNS-1652544 and CNS-1505701 grants, and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy. This material is also based on research sponsored by the ONR under agreements number N00014-17-1-2012 and N00014-17-1-2504.

dynamical systems, our idea is somewhat complementary to the self/event-triggered control paradigm that is used to reduce network utilization in networked control systems [13].

We introduce a method to relate the QoC guarantees in the presence of attack and the bandwidth overhead due to the use of intermittent data authentication in communication over a shared network. This lays the foundation for our network schedulability analysis for non-preemptive real-time sensor messages with intermittent integrity enforcements, which ensure predefined QoC requirements. Furthermore, we present a Mixed-Integer Linear Programming (MILP)-based technique to perform tradeoff analysis between the network utilization and the overall QoC guarantees for a set of control loops communicating over a shared network. This facilitates optimal bandwidth allocation that maximizes the overall QoC guarantees in the presence of attacks with respect to the available resources (i.e., network bandwidth). Finally, on a real-world automotive case study, we illustrate how the proposed design-time framework can be used to provide secure-control guarantees for CAN-based CPS. Specifically, we show how we can integrate sensor measurements with intermittent data authentication such that we maximize QoC under attack while ensuring that timing guarantees for existing real-time messages are not violated.

This paper is organized as follows. Section II introduces the problem considered in this work, before we present the concept of intermittent data integrity enforcements and a framework to relate QoC guarantees in the presence of attacks with the integrity enforcement rate (Section III). In Section IV, we present a real-time message model with intermittent authentication, and in Section V we introduce an MILP-based method for synthesis of schedulable messages with QoC guarantees based on intermittent data authentication. Section VI presents a method to derive a message set with the optimal balance between the overall QoC and available resources. Finally, in VII we evaluate our approach on a real-world automotive case study, before providing concluding remarks in Section VIII.

II. Motivation and Problem Statement

In this work, we focus on networked control of N discrete-time plants Σ_i, $i = 1, ..., N$, of the form

$$\mathbf{x}_i[k+1] = \mathbf{A}_i\mathbf{x}_i[k] + \mathbf{B}_i\mathbf{u}_i[k] + \mathbf{w}_i[k],$$
$$\mathbf{y}_i[k] = \mathbf{C}_i\mathbf{x}_i[k] + \mathbf{v}_i[k],$$

where $\mathbf{x}_i[k] \in \mathbb{R}^n$, $\mathbf{u}_i[k] \in \mathbb{R}^m$, and $\mathbf{y}_i[k] \in \mathbb{R}^p$ denote state, input and outputs of the i^{th} plant at time k, respectively, and $\mathbf{w}_i \in \mathbb{R}^n$ and $\mathbf{v}_i \in \mathbb{R}^p$ are the process and measurement noise. The above models are obtained by discretizing the corresponding continuous plant model. Although we assume that $\mathbf{w}_i$ and $\mathbf{v}_i$ are independent identically distributed Gaussian random variables our work can be extended to bounded-size noise.

For each system Σ_i, we specify designed controllers as

$$\hat{\mathbf{x}}_i[k+1] = \mathbf{f}_i\left(\hat{\mathbf{x}}_i[k], \hat{\mathbf{y}}_i[k]\right),$$
$$\mathbf{u}_i[k] = \mathbf{g}_i\left(\hat{\mathbf{x}}_i[k], \hat{\mathbf{y}}_i[k]\right),$$

where $\mathbf{f}_i(\cdot)$ and $\mathbf{g}_i(\cdot)$ are any linear mappings, $\hat{\mathbf{x}}_i[k]$ is the controller's state, such as the estimated plant state, and $\hat{\mathbf{y}}_i[k]$ are received sensor measurements in step k. The above formulation is general, capturing observer-based state feedback as well as standard feedback controllers, which can be designed using various techniques focused on stability, optimal performance or robustness to modeling errors.

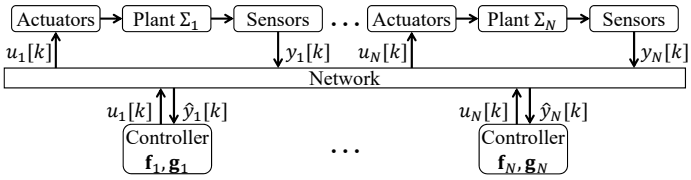

Fig. 1. Networked system architecture for with N control-loops; note that in general some controllers may be mapped on shared CPUs.

Sensor measurements are transmitted as real-time messages over a network illustrated in Fig. 1, which is shared with non real-time communication packets. We abstract a standard real-time message by a 3-tuple $M(c, p, d)$ where c is the *non-preemptive* message transmission time, p is the transmission period – i.e., the time between message arrivals (equal to the plant's sampling time), and d is the message deadline relative to its arrival time. To simplify our notation, non real-time messages are abstracted with a single parameter c_{max}^{NRT} that captures the transmission time of the longest such message.

To model the attacker, we use the standard attack model from [9]–[11], [14]. When no MitM attacks on the network occur, we have that $\hat{\mathbf{y}}_i[k] = \mathbf{y}_i[k]$. On the other hand, with MitM attacks, sensor measurements received by the controller $\hat{\mathbf{y}}_i[k]$ could potentially differ from the actual sensor measurements $\mathbf{y}_i[k]$. We differentiate system evolutions with and without attacks by adding superscript a to all variables affected by the attacker's influence. For example, the plant's state and outputs when the system is under attack are denoted as $\mathbf{x}_i^a[k]$ and $\mathbf{y}_i^a[k]$, respectively. Thus, attacks on sensor measurements delivered to the controller can be modeled as

$$\hat{\mathbf{y}}_i^a[k] = \mathbf{y}_i^a[k] + \mathbf{a}_i[k] = \mathbf{C}_i\mathbf{x}_i^a[k] + \mathbf{v}_i^a[k] + \mathbf{a}_i[k],$$

where $\mathbf{a}_i[k]$ is a sparse vector capturing values injected by the attacker. Note that sparsity of vector $\mathbf{a}_i[k]$ depends on the set of compromised sensor flows – if communication from a sensor to the controller is not corrupted then the corresponding value in $\mathbf{a}_i[k]$ has to be zero. Hence, we can capture any assumptions about the set of compromised sensor flows (e.g., the number of the flows) by introducing constraints on the sparsity of the vector. Yet, unless stated otherwise, we simplify our presentation by focusing on the worst-case scenario, where the attacker can compromise all sensor flows once he decides to launch an attack.

Commonly, MitM attacks are dealt with by employing standard cryptographic mechanisms such as MACs; we assume that the attacker does not have access to the shared secret keys used to generate the MACs. Thus, when authentication is enforced with the use of MACs, we assume that the attacker avoids inserting false data measurements in order to stay undetected, meaning that at these times $\mathbf{a}_i[k] = \mathbf{0}$.[1] We assume that the attacker has full knowledge of the system, enabling him to smartly craft false measurements in order to deceive the controller into pushing the plant away from the desired

[1] Although the attacker could potentially prevent authenticated messages from being delivered, we do not consider such attacks, since Denial-of-Service attacks are easier to detect in CPS with reliable communication networks.

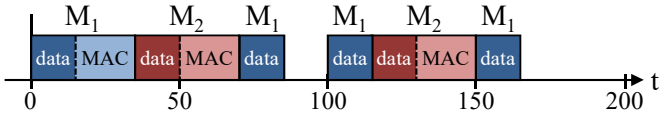

Fig. 2. Scheduling messages from two sensors; feasible message set $M_1(15, 50, 50)$ and $M_2(15, 100, 100)$ becomes infeasible when MACs of length 20 are added to every message. However, they can be scheduled if e.g., every fourth transmission of M_1 is authenticated, while M_2 is authenticated on every period.

operating point. The attacker also knows the times when authentication will be used, allowing him to plan ahead and avoid being detected. Finally, the attacker's goal is to maximally reduce control performance (i.e., QoC), using the inserted false measurements, while remaining stealthy – i.e., undetected by the system; thus, in addition to not inserting false data packets in time-frames when authentication is enforced, the falsified sensor measurements should not trigger the Intrusion Detection System (IDS) employed at the controller.

However, ensuring authentication for every transmitted sensor measurement could impose unfeasible constraints on the underlying network. For example, consider two periodic real-time sensor messages modeled as $M_1(15, 50, 50)$ and $M_2(15, 100, 100)$ when MACs are not added. These two messages can be scheduled over the network. However, if adding MACs increases transmission time for M_1 and M_2 by 20 time units, the resulting message set $M_1'(35, 50, 50)$ and $M_2'(35, 100, 100)$ becomes unfeasible. On the other hand, if every fourth message for M_1 is authenticated, the messages can meet their deadlines as illustrated in Fig. 2. From the perspective of QoC guarantees even with the adversarial presence, this level of integrity guarantees may be sufficient; we recently showed that even intermittent data integrity guarantees significantly limit the attacker's impact [11], [12].

Therefore, this work focuses on tradeoffs between the QoC in the presence of attacks and integrity enforcement overhead for sensor messages. We address the following problems:

- How to map requirements for QoC in the presence of attacks into authentication constraints for real-time sensor messages?
- How can such real-time messages be scheduled over a shared network, while ensuring the desired QoC level for each of the control loops even in the presence of attacks?
- How to perform optimal bandwidth allocation for each control loop such that the overall (i.e., for all loops) security guarantees, in terms of QoC under attacks, are maximized?

We start with our recently introduced framework for security aware control with intermittent data-integrity enforcements.

III. Security-Aware Control with Intermittent Data Integrity Enforcements

CPS controllers usually incorporate a state estimator feeding a feedback controller as shown in Fig. 3. Furthermore, an IDS is used to detect discrepancies between physical properties of the system (i.e., its model) and the received sensor measurements. The actual IDS employed in these application directly depends on the plant (specifically noise) model. For example, for bounded-size noise, security-aware estimators and set-based IDSs have been recently proposed (e.g., [14]). Similarly, for Gaussian noise model, Kalman filter-based estimators can

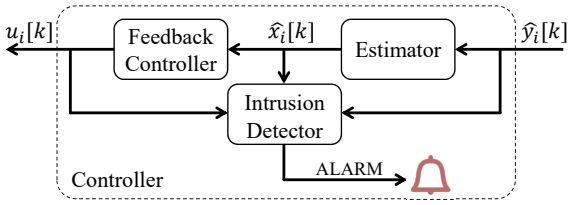

Fig. 3. General controller architecture.

be used with statistical IDSs, such as χ^2 [9]–[11] or Sequential Probability Ratio Test (SPRT) detectors [12].

It was recently shown (e.g., [10], [14]) that a stealthy attack can significantly reduce QoC when the attacker is able to compromise a certain number of sensor flows. For any type of the considered controllers (i.e., estimators and IDSs), this can be achieved by injecting false sensor measurements that result in a skewed state estimation; this in-turn deceives the controller into steering the system away from the desired trajectory/operating point by applying ill-suited control commands. However, the state estimation error has to be slowly increased in order for the attacker to stay undetected. This, coupled with the fact that each plant has its own dominant time-constant (captured by the plant model Σ_i) implies that QoC can be significantly degraded only some time after a stealthy attack is launched.

To analyze this formally, we introduce the reachable region $\mathcal{R}[k]$ of the state estimation error under attack (i.e., $\mathbf{e}^a[k]$), k steps after the attack is launched. For plants with Gaussian noise, as in this work, the regions can be defined as [11], [12][2]

$$\mathcal{R}[k] = \left\{ \mathbf{e} \in \mathbb{R}^n \;\middle|\; \begin{array}{c} \mathbf{e}\mathbf{e}^T \preccurlyeq E[\mathbf{e}^a[k]]E[\mathbf{e}^a[k]]^T + \gamma Cov(\mathbf{e}_k^a), \\ \mathbf{e}^a[k] = \mathbf{e}_k^a(\mathbf{a}_{1..k}),\ \mathbf{a}_{1..k} \in \mathcal{A}_k \end{array} \right\}.$$

Here, $\mathbf{a}_{1..k} = \left[\mathbf{a}[1]^T ... \mathbf{a}[k]^T\right]^T$ captures all injected false sensor measurements, $\mathcal{A}_k$ denotes the set of all stealthy attacks, and $\mathbf{e}_k^a(\mathbf{a}_{1..k})$ is the estimation error evolution caused by the attacks $\mathbf{a}_{1..k}$. In [12], we also showed that $Cov(\mathbf{e}_k^a)$ is equal to the estimation error covariance matrix when no attacks are introduced, and thus is known in advance. Furthermore, the global reachable region $\mathcal{R}$ (i.e., for all $k > 0$) of the state estimation error $\mathbf{e}^a[k]$ is the set $\mathcal{R} = \bigcup_{k=0}^{\infty} \mathcal{R}[k]$.

In [11], [12], we recently introduced techniques to tightly evaluate regions $\mathcal{R}[k]$, starting from the system model (i.e., plant dynamics Σ_i and employed IDS) as well as the attack model from Section II, which can be extended with additional potentially available information, including the maximal number of compromised sensor flows; when such information is not available, we assume that measurements from all sensors can be compromised. In addition, these techniques facilitate capturing the effects of data integrity enforcements at specific time-points defined by integrity enforcement policy μ.

Definition 1 ([11], [12]): Intermittent data integrity enforcement policy (μ, l), where $\mu = \{t_k\}_{k=0}^{\infty}$, with $t_{k-1} < t_k$ for all $k > 0$ and $l = \sup_{k>0} t_k - t_{k-1}$, ensures that $\mathbf{a}_{t_k} = \mathbf{0}$, for all $k \geq 0$.

Definition 1 imposes a maximum time between integrity enforcements, captured by the parameter l. It also captures periodic enforcements when $l = t_k - t_{k-1}$ for all $k > 0$, as well

[2]A similar definition can be used for systems with bounded-size noise [14].

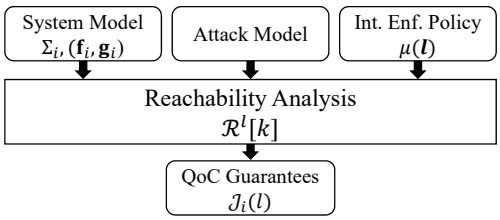

Fig. 4. Design-time framework to evaluate effects of intermittent integrity enforcement policies on QoC guarantees in the presence of attack based on the reachability analysis from [11], [12].

as policies with continuous integrity enforcements (for $l = 1$). Since our goal is to reduce communication overhead associated with integrity enforcement, we will initially focus on policies where enforcements are maximally spread apart, i.e., for which $l = t_k - t_{k-1}$ for all $k > 0$.

In general, QoC depends on state estimation errors. For instance, as illustrated in [15] when linear-quadratic control cost is consider as QoC, a tight bound on QoC degradation can be obtained as $\mathcal{L}_2$-gain (which is known in advance) scaled bound on the size of estimation error. Consequently, tight guarantees on the size of state-estimation error due to attacks can be utilized to capture QoC in the presence of attacks. This effectively allows us to obtain a design-time reachability-based framework from Fig. 4 to evaluate impact of stealthy attacks on systems with (and without) integrity enforcement policies. Furthermore, the system and attack models are fixed for any CPS under consideration, and therefore the framework can be used to analyze impact of the integrity enforcement parameter l on the attack-induced state estimation error (and thus QoC). Formally, this can be captured using $\mathcal{J}(l)$ functions defined as

$$\mathcal{J}(l) = supp\{\|\mathbf{e}\|_2 \;\mid\; \mathbf{e} \in \mathcal{R}^l\}, \quad \text{where} \quad \mathcal{R}^l = \bigcup_{k=0}^{\infty} \mathcal{R}^l[k],$$

and $\mathcal{R}^l[k]$ denotes $\mathcal{R}[k]$ computed for all integrity policies with parameter l. For example, functions $\mathcal{J}_i(l)$ for three automotive closed-loop systems are presented in Fig. 8.

The aforementioned $\mathcal{J}_i(l)$ functions are the foundation for our analysis of tradeoffs between QoC guarantees in the presence of attacks and the required network resources employed for data authentication. In addition, since $\mathcal{J}_i(l)$ are non-decreasing functions of l, for each plant Σ_i, QoC requirements (e.g., a bound on $\mathcal{J}_i(l)$) can be mapped into constraints on l_i – i.e., the number of non-authenticated communication packets between consecutive authenticated ones.

We show effects of integrity enforcements on automotive cruise control by focusing on the reachable regions for state estimation errors (Fig. 5); the vehicle can be modeled as a dynamical system [16] with three states capturing the difference between the desired and current distance from the preceding vehicle (x_1), the difference between the desired and developed speed (x_2), and acceleration (x_3). A stealthy attack that compromises only distance measurements can still result in unbounded estimation errors when no data integrity is enforced. On the other hand, when distance sensor's integrity is enforced at time $k = 4$, there exists a notable reduction in the size of 4-reachable region for estimation error.

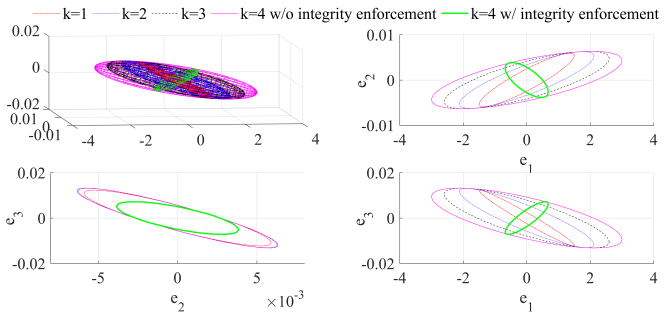

Fig. 5. Evolution of the state estimation error regions $\mathcal{R}[k]$ for automotive cruise control in the presence of attacks on the distance sensor, with and without data integrity enforcement at $k = 4$. Estimation errors for states that correspond to the distance, speed, and acceleration are e_1, e_2, and e_3, respectively; since $\mathcal{R}[k] \in \mathbb{R}^3$, corresponding 2D projections are also presented.

IV. Modeling of Real-Time Messages with Intermittent Authentication

Let's revisit the example from Section II with two periodic real-time messages $M_1(15, 50, 50)$ and $M_2(15, 100, 100)$, as well as the corresponding messages $M_1'(35, 50, 50)$ and $M_2'(35, 100, 100)$ when authentication is added. Since network utilization for each message M_i and the overall utilization are defined as $U_i = \frac{c_i}{p_i}$ and $U_{\mathcal{M}} = \sum_{i=1}^{N} U_i$, respectively, it follows that the set of messages with continuous data authentication has $U_{\mathcal{M}'} = 1.05$, and is thus infeasible.

On the other hand, assume for example, that integrity enforcement for data transmitted via M_1 is required only every fourth transmission (i.e., every 200 time units). Then the network demand for these two messages can be depicted as shown in Fig. 6(a), (b). Now, let's assume that both authenticated transmissions of M_1 and M_2 are ready at $t = 0$ when the network has just started transmitting a non real-time message of length 25 time units. In this case, M_1 will miss its deadline at $t = 50$, as shown in Fig. 6(c). However, if the initial authentication of message M_1 is delayed by, for example, 100 time units, the messages are schedulable with EDF scheduler, as shown in Fig. 6(d). Furthermore, note that the integrity requirements are not violated since every sequence of four consecutive transmissions of message M_1 contains exactly one authenticated transmission.

As illustrated in the example, it is beneficial to expand the standard real-time message model by allowing for periodic message extensions that include a MAC. Additionally, to give a degree of freedom during scheduling and avoid the scenario from Fig. 6(c), the model should facilitate capturing offsets to the initial authentication. Thus, we model the set $\mathcal{M}$ of real-time messages with intermittent authentication by defining each message as $M_i(C_i, p_i, l_i, s_i)$, $1 \leq i \leq N$ where

- $C_i = [c_i^{norm}, c_i^{ext}]$ contains the transmission times, normal and extended, of the i^{th} message in non-authenticated and authenticated transmission mode, respectively,
- p_i is the normal message period – i.e., the time between consecutive message transmission requests,
- l_i is the period of extended messages specified as an integer multiple of normal message periods – i.e., every l_i consecutive messages contain exactly one authenti-

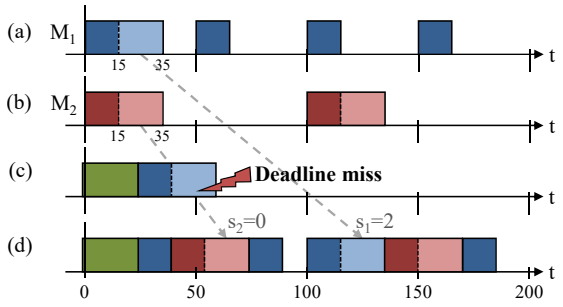

Fig. 6. Two messages $M_1([15,35],50,4,s_1)$ and $M_2([15,35],100,1,s_2)$ sharing a network with a non real-time message with transmission time 25 time units. As shown in (c) this message set is infeasible if both messages are authenticated at $t=0$ (deadline miss at $t=50$). However, if the initial authenticated transmission of M_1 is offset by, for example, 100 time units ($s_1=2$), this message set becomes feasible, as shown in (d), while integrity enforcement requirements remain satisfied.

cated message,

- s_i is the offset of the initial authenticated message that satisfies $0 \leq s_i \leq l_i - 1$, i.e., the transmission request time of the first authenticated message is $s_i p_i$.

To simplify our notation we assume that the relative deadline of each message M_i is p_i, although this work can be directly extended to cover any deadline $d_i \leq p_i$, which would in turn facilitate capturing of local packet processing and control updating tasks at each CPU. Finally, for the message model, the message and overall utilizations are

$$U_i = \frac{c_i^{norm}}{p_i} + \frac{c_i^{ext} - c_i^{norm}}{l_i p_i}, \qquad U_{\mathcal{M}} = \sum_{i=1}^{N} U_i. \quad (1)$$

Given the presented message model, we pose two essential problems. First, note that in our example from Fig. 6, offsets of the initial authenticated transmissions were not a priori given. In fact, our goal is to determine a set of offsets of initial authenticated messages $s_1, ..., s_N$, if such set exists, that yields a feasible set of messages over a shared network, while still satisfying integrity enforcement requirements captured as predefined $l_1, ..., l_N$. Furthermore, a closer inspection of our example in Fig. 6, yields to a conclusion that integrity can be enforced in every third transmitted message for M_1 (i.e., $l_1 = 3$), instead of every fourth one, while still ensuring network schedulability. This would effectively improve QoC guarantees in the presence of attacks, as described in Section II. Therefore, the second problem can be cast as an optimization problem that strives to find an assignment of initial authenticated transmission offsets $(s_1, ..., s_N)$ and integrity enforcement rates $(l_1, ..., l_N)$ that minimize the overall QoC degradation while ensuring network schedulability, and even allocating some level of utilization for non real-time messages. Here, the overall QoC degradation can be captured as $\sum_{i=1}^{N} \omega_i \mathcal{J}_i(l_i)$, where weights $\omega_i > 0$ encode the importance of the specific control loop. Note that this must be done with respect to the minimum required integrity enforcement rate (captured by $l_1^{max}, ..., l_N^{max}$), that guarantees the minimum QoC specified at design time. The aforementioned problems can be formally specified as follows.

Problem 1: For a set of real-time messages $\mathcal{M}$ with $l_1, ..., l_N$ capturing prespecified QoC requirements, find offsets $s_1, ..., s_N$ for initial authenticated messages such that the obtained complete set $\mathcal{M}$ is feasible under non-preemptive EDF.

Problem 2: For a set of real-time messages $\mathcal{M}$ and a set of associated cost functions $\mathcal{J}_i(l_i)$, $i = 1, ..., N$, find offsets $s_1, ..., s_N$ for initial authenticated messages and optimal authentication periods $l_1, ..., l_N$ such that the obtained complete message set $\mathcal{M}$ is feasible under non-preemptive EDF, and the objective $\sum_{i=1}^{N} \omega_i \mathcal{J}_i(l_i)$ is minimized.

Finally, it is important to highlight that we focus on schedulability with EDF scheduler, which is optimal for non-idle schedules and it outperforms rate-monotonic schedulers for realistic loads on networks such as CAN [13], [17].

V. Scheduling QoC-Aware Network Messages with Intermittent Authentication

In this section, we introduce a method to solve Problem 1. Specifically, we start with schedulability conditions for non-premeptive messages under EDF, before presenting a MILP formulation to obtain feasible authentication offsets $s_1, ..., s_N$.

A. Schedulability with Non-Preemptive EDF

Schedulability conditions for a set of standard real-time messages under non-preemptive EDF were introduced in [18].

Theorem 1 ([18]): Consider a set of real-time messages $M_i(c_i, p_i, d_i)$, $1 \leq i \leq N$. The message set is schedulable under non-preemptive EDF over a network shared with non real-time messages with maximum transmission time c_{max}^{NRT} if $\sum_{i=1}^{N} \frac{c_i}{p_i} \leq 1$ and

$$\sum_{i=1}^{N} max\left\{0, \left\lfloor \frac{t - d_i}{p_i} \right\rfloor + 1\right\} c_i + c_m \leq t_k, \forall t_k \in TS, \quad (2)$$

where $TS = \bigcup_{i=1}^{N} \left\{ d_i + jp_i | j = 0, ..., \left\lfloor \frac{t_{max} - d_i}{p_i} \right\rfloor \right\}$, $t_{max} = \max\left\{ d_1, ..., d_N, \left(c_m + \sum_{i=1}^{N} \left(1 - \frac{d_i}{p_i}\right) c_i \right) / (1 - U_{\mathcal{M}}) \right\}$, and $c_m = max\{c_{max}^{NRT}, \max_{i=1}^{N} c_i\}$.

Remark 1: The bound on the time testing set in Theorem 1 depends on t_{max}, which significantly increases as utilization approaches one. In [18], the authors suggest that highly utilized links/networks should be avoided. Another option is to test the above condition (2) over the whole hyperperiod P_H, beyond which the schedule repeats, where $P_H = lcm\{p_1, ..., p_N\}$, i.e., the least common multiple of message periods $p_1, ..., p_N$.

Remark 2: The condition in (2) does not support offsets, as it was derived for sporadic messages. Consequently, it necessarily defends against the worst-case message alignment. Thus, it can sufficiently be used for periodic messages with offsets as long as offsets are integer multiples of periods, since this shifting of messages does not introduce any new arrival patterns. However, the condition in (2) must then be evaluated at absolute deadlines and t_{max} must be extended accordingly.

Observe that the network load condition from (2), takes the form of a weighted sum, where weight factors are message transmission times and weighted addends are integers counting the number of transmissions of every message, from the zeroth instant and up to time t_k. We capture these message counts

contributing to the network demand during the time $[0, t_k]$ as

$$\eta_i^{n\&e}(t_k) = max\left\{0, \left\lfloor \frac{t_k - p_i}{p_i} \right\rfloor + 1\right\} \quad (3)$$

for the number of normal and extended messages, and

$$\eta_i^{ext}(t_k) = max\left\{0, \left\lfloor \frac{t_k - p_i - s_i p_i}{l_i p_i} \right\rfloor + 1\right\} \quad (4)$$

for the number of extended messages (with MAC). This allows us to formulate the total link demand of real-time messages $M_i(C_i, p_i, l_i, s_i)$, $1 \leq i \leq N$ up to the time t_k as

$$\sum_{i=1}^{N} \left(\eta_i^{c\&e}(t_k) c_i^{norm} + \eta_i^{ext}(t_k)\Delta c_i\right).$$

The time testing set in our case has to include deadlines of all (normal and extended) messages, which are multiples of normal message periods p_i. The upper bound on the time testing instants becomes $\left\lfloor \frac{t_{max} - p_i}{p_i} \right\rfloor$ given that $d_i = p_i$ and normal frame offsets are always zero. Similarly, t_{max} takes the maximum value of all offset deadlines (due to extended messages), and its upper bound that depends on the utilization (i.e., $(c_m + \sum_{i=1}^{N}(1 - \frac{d_i}{p_i})c_i)/(1 - U_{\mathcal{M}})$ in Theorem 1), transforms into $(c_m + \sum_{i=1}^{N} \frac{l_i - 1}{l_i} c_i^{ext})/(1 - U_{\mathcal{M}})$. This is caused by the fact that only extended messages yield a non-zero numerator in the sum – i.e., the period of these messages is $l_i p_i$ while deadlines are still p_i and transmission times of these messages are c_i^{ext}. Finally, c_m remains the transmission time of the longest of all messages. Thus, we can formulate the following result.

Theorem 2: A set of real-time messages $M_i(C_i, p_i, l_i, s_i)$, $i \leq N$ is schedulable by non-preemptive EDF if $U_{\mathcal{M}} \leq 1$ and

$$\sum_{i=1}^{N} \left(\eta_i^{c\&e}(t_k) c_i^{norm} + \eta_i^{ext}(t_k)\Delta c_i\right) + c_m \leq t_k, \ \forall t_k \in TS, \quad (5)$$

where $TS = \bigcup_{i=1}^{N} \left\{ jp_i | j = 1, ..., \left\lfloor \frac{t_{max} - p_i}{p_i} \right\rfloor \right\}$, $t_{max} = \max\left\{ \max_{i=1}^{N}(s_i + 1)p_i, \frac{c_m + \sum_{i=1}^{N} \frac{l_i - 1}{l_i} c_i^{ext}}{1 - U_{\mathcal{M}}} \right\}$, $\Delta c_i = c_i^{ext} - c_i^{norm}$, and $c_m = max\{c_{max}^{NRT}, \max_{i=1}^{N} c_i^{ext}\}$.

Proof: The proof follows directly from the proof of the corresponding theorem from [18] and is thus omitted. ∎

B. Message Set Completion with Predefined QoC

For synthesis of feasible message sets (i.e., to solve Problem 1) our goal is to find a set of parameters $s_1, ..., s_N$ resulting in a complete message set that is feasible under non-preemptive EDF. In other words, we consider our message sets as incomplete, i.e., offsets of initial authenticated transmissions are taken to be variables. On the other hand, QoC requirements are predefined as specific integrity enforcement rates $l_1, ..., l_N$.

Let us define binary variables $a_{k,j}^i$ as indicators that by the k^{th} instant the j^{th} authenticated transmission of message M_i should have completed transmission. Here, $1 \leq i \leq N$, $1 \leq j \leq \left\lfloor \frac{t_{max}}{l_i p_i} \right\rfloor$, $1 \leq k \leq |TS|$. The relation between variables $a_{k,j}^i$ and real-time message parameters can be expressed as

$$a_{k,j}^i = 1 \Leftrightarrow t_k \geq (s_i + 1)p_i + (j - 1)l_i p_i. \quad (6)$$

For example, for the schedule from Fig. 6(d), variable assignment for message M_1 is $a_{1,1}^1 = 0, a_{2,1}^1 = 0, a_{3,1}^1 = 1, a_{4,1}^1 = 1$. From (4) it follows that $\eta_i^{ext}(t_k) = \sum_{j=1}^{\left\lfloor \frac{t_{max}}{l_i p_i} \right\rfloor} a_{k,j}^i$, while $\eta_i^{c\&e}(t_k)$ from (3) evaluates to a constant for any time instant. Thus, we can express the condition (5) as

$$\sum_{i=1}^{N} \left(c_i^{norm} \eta_i^{c\&e}(t_k) + \Delta c_i \sum_{j=1}^{\left\lfloor \frac{t_{max}}{l_i p_i} \right\rfloor} a_{k,j}^i \right) + c_m \leq t_k, \quad (7)$$

with $c_m = max\{c_{max}^{NRT}, \max_{i=1}^{N} c_i^{ext}\}$. The network demand condition is now expressed as a set of linear constraints since it only depends on binary variables $a_{k,j}^i$. The logical conditions from (6) can be cast as linear constraints by applying the "Big M" method [19]. Thus, equivalent linear constraints are

$$(s_i + 1)p_i + (j - 1)l_i p_i \leq t_k + M(1 - a_{k,j}^i), \quad (8)$$

$$(s_i + 1)p_i + (j - 1)l_i p_i > t_k - M a_{k,j}^i, \quad (9)$$

where M is a large constant. Also, for integrity requirements to be satisfied (i.e., that in every l_i transmission periods, exactly one transmission is authenticated), integer variables s_i satisfy

$$0 \leq s_i \leq l_i - 1. \quad (10)$$

Finally, the complete MILP formulation that finds a feasible solution to Problem 1 consists of constraints (7)-(10), where indices are in their respective ranges — i.e.,

$$1 \leq i \leq N, \qquad 1 \leq j \leq \left\lfloor \frac{t_{max}}{l_i p_i} \right\rfloor, \qquad 1 \leq k \leq |TS|. \quad (11)$$

Note that since only feasibility is of interest here, we did not specify any objectives for the optimization problem above. In addition, the obtained assignment $s_1, ..., s_n$, if existent, produces a feasible message set due to Theorem 2.

Finally, since MILP solvers require the use of non-strict inequalities, (9) can be expressed as

$$(s_i + 1)p_i + (j - 1)l_i p_i \geq t_k - M a_{k,j}^i + \varepsilon,$$

for a small $\varepsilon > 0$. In this case, the values for M and ε have to be assigned in a way that assures no potential errors are introduced due to finite precision implementations of MILP solvers. More details can be found in [20, Remark 1].

VI. QoC-Optimal Bandwidth Allocation

To solve Problem 2, optimal link allocation is of interest, i.e., we wish to increase the integrity enforcement rates as much as possible while maintaining schedulability. This exploits the fact that QoC degradation functions $\mathcal{J}_i(l_i)$ map the integrity enforcement rate into QoC. In addition, with respect to the lowest allowable QoC for a given control loop, minimum integrity enforcement rates are defined through a set of constants $l_1^{max}, ..., l_N^{max}$.

We identify a couple of challenges in solving Problem 2. First, suitable cost functions need to be formed capturing the relationship between the integrity enforcement rate and QoC, in a way that supports solving the optimization problem. Based on a weighted sum of these functions, we can optimize the overall QoC subject to schedulability constraints. Second, it is necessary to avoid specifying the exact utilization in the bound of the time testing set t_{max} in (5), as the overall utilization

$U_{\mathcal{M}}$ it is not known while optimizing authentication rates. We address these challenges in the remainder of this section.

Message parameters $l_1, ..., l_N$ are now variables bounded from above with the minimum QoC requirement $1 \leq l_i \leq l_i^{max}$, $1 \leq i \leq N$. This does not affect the linearity of the problem and constraints expressed in (7)-(10) remain unchanged. The first challenge to address is specifying QoC degradation functions $\mathcal{J}_i(l_i)$. As we discussed in Section III, our reachability analysis framework provides numerical descriptions for these functions. We observe that for practical systems, piecewise-linear approximations can be fitted to QoC degradation functions $\mathcal{J}_i(l_i)$ without significant effects on accuracy, as shown on example cost functions in Fig. 8. Therefore, we can adopt the piece-wise linear description of approximated QoC degradation functions as

$$\hat{\mathcal{J}}_i(l_i) = \sum_{r=1}^{F_i} \left((\alpha_r^i l_i + \beta_r^i) b_r^i \right).$$

Here, F_i denotes the number of approximating linear segments of the cost function for the i^{th} closed-loop, $\alpha_r^i l_i + \beta_r^i$ is the equation of the r^{th} segment over the range $l_i \in [\underline{l}_i^r, \bar{l}_i^r]$, and $\hat{\mathcal{J}}_i(l_i)$ is continuous so that $[1, l_i^{max}] = \bigcup_{r=1}^{F_i} [\underline{l}_i^r, \bar{l}_i^r]$. Selector variables $b_r^i \in \{0, 1\}$ ensure that the correct linear segment is enabled based on the current value of l_i – i.e.,

$$b_r^i = 1 \quad \Rightarrow \quad \underline{l}_i^r \leq l_i \leq \bar{l}_i^r, \quad 1 \leq r \leq F_i. \tag{12}$$

For example, the QoC-degradation curve in Fig. 8(left) has 4 segments: $\{[1,2],[2,3],[3,18],[18,30]\}$. Note that the multiplication of variables $b_r^i l_i$ is nonlinear. This can be solved by introducing additional variables $c_r^i = b_r^i l_i$. By applying the "Big M" method, as before, a set of linear constraints specifying these relations can be formulated as

$$\begin{aligned} \underline{l}_i^r - M(1 - b_r^i) \leq l_i \leq \underline{l}_i^r + M b_r^i, \\ \bar{l}_i^r - M b_r^i \leq l_i \leq \bar{l}_i^r + M(1 - b_r^i), \end{aligned} \tag{13}$$

$$\sum_{r=1}^{F_i} b_r^i = 1, \qquad 1 \leq i \leq N, \tag{14}$$

$$c_r^i \leq b_r^i M, \qquad c_r^i \geq l_i - (1 - b_r^i) M, \qquad 0 \leq c_r^i \leq l_i. \tag{15}$$

Here, constraints (13) implement selector variable conditions in (12). Constraints in (14) guarantee that exactly one linear segment of the piecewise linear approximation is active. The first constraint in (15) provides $c_r^i = 0$ when the corresponding segment is inactive, i.e., $l_i \notin [\underline{l}_i^r, \bar{l}_i^r]$ and $b_r^i = 0$, while the second one guarantees $c_r^i = 1$ when $l_i \in [\underline{l}_i^r, \bar{l}_i^r]$ and $b_r^i = 1$.

Note that the number of authenticated transmissions, captured as the range of j in (11), must account for the case when every message authenticates every transmission, since the range of indices in MILP may not depend on variables. Thus, we consider the upper bound for the number of authenticated transmissions j as $\left\lfloor \frac{t_{max}}{l_i p_i} \right\rfloor_{l_i^{min}=1} = \left\lfloor \frac{t_{max}}{p_i} \right\rfloor$. Additional issue arises from the need to specify exact link utilization while calculating the bound on the time testing set in (5), because $U_{\mathcal{M}}$ affects t_{max}. One solution is to set an upper bound on the overall utilization. This can be of practical use since the network is usually shared between real-time and non real-time messages. In this scenario, the system designer can designate

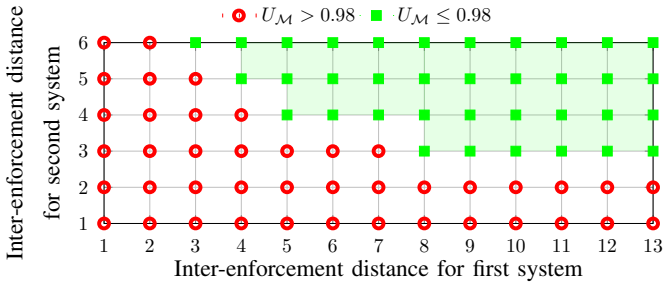

Fig. 7. Example of an allowable region of inter-enforcement distances given a preset upper utilization bound of $\bar{U}_{\mathcal{M}} = 0.98$, for the cruise control and steering control for lane tracking case studies from Section VII.

a lower-bound on utilization assigned to non real-time traffic.

We can transform the expression for utilization in (1) into an inequality specifying upper link utilization bound as

$$U_{\mathcal{M}}(l_1, ..., l_N) = \sum_{i=1}^{N} \frac{\Delta c_i}{l_i p_i} + \sum_{i=1}^{N} \frac{c_i^{norm}}{p_i} \leq \bar{U}_{\mathcal{M}}, \tag{16}$$

where $\bar{U}_{\mathcal{M}}$ is the desired upper bound for network utilization. Notice that integrity enforcement rates for all messages are encoded in variables l_i, and those are the only variables in (16). The relation in (16) is, however, not linear. Moreover, a seemingly convenient change of variables $f_i = \frac{1}{l_i}$ as in [21] results in constraints that are not easily cast as linear. We thus take a different approach by directly capturing the set of integer values $l_1, ..., l_N$ for which $U_{\mathcal{M}}(l_1, ..., l_N)$ satisfies (16). For example, Fig. 7 illustrates an example of such a region when integrity enforcement is optimized for two systems and a predefined upper utilization bound, based on a realistic case study from Section VII. Here, it is important to note that this region can be expressed using linear constrains on variables l_1 and l_2 as: $l_1 \geq l_1^{min} = 3$, $l_2 \geq l_2^{min} = 3, l_1 + l_2 \geq 9$, $M(l_2 - 3) \geq 8 - l_1$. The first two constraints bound variables l_1 and l_2 from below, and the third ensures the variables do not take any values below the line $l_1 + l_2 = 9$. Since this is not sufficient (e.g., points $(l_1, l_2) \in \{(6,3),(7,3)\}$ satisfy the three constraints but violate the utilization bound), the last constraint ensures that $l_2 = 3 \Rightarrow l_1 \geq 8$ holds.

Even though the added number of constraints is not significant, this specification of an upper utilization bound becomes less appealing when the number of messages with optimizable integrity enforcement rate rises, and as the allowable region gets more complex. Nevertheless, this technique offers an efficient way of completing a QoC-optimal message set.

Consequently, we can now specify our MILP-formulation for the QoC-optimal link allocation as the following problem

$$\begin{aligned} \min \sum_{i=1}^{N} \left[\omega_i \sum_{r=1}^{F_i} (\alpha_r^i c_r^i + \beta_r^i b_r^i) \right] \\ \text{subject to: (7)-(10), (13)-(15),} \\ l_i^{min} \leq l_i \leq l_i^{max}, \qquad U_{\mathcal{M}}(l_1, ..., l_N) \leq \bar{U}_{\mathcal{M}} \\ 1 \leq i \leq N, \qquad 1 \leq j \leq \left\lfloor \frac{t_{max}}{p_i} \right\rfloor, \qquad 1 \leq k \leq |TS|. \end{aligned} \tag{17}$$

In the case of our running example, this formulation produces a total of 92 variables, six of which are integers (s_i-s,

l_i-s, c_r^i-s), and a total of 193 constraints if a two-segment cost function is used for M_1. Resulting integrity enforcement rates are, as previously expected, $l_1^* = 3$, $l_2^* = 1$, if maximum rates are set at initial values of the example ($l_1^{max} = 4$, $l_2^{max} = 1$).

The following section proposes an approach to deal with an a-priori unknown utilization for optimal link allocation.

A. Opportunistic Bandwidth Allocation

Consider a set $\mathcal{M}$ of complete real-time messages $M_i, i = 1, ...N$; we assume that offsets of initial authenticated transmissions $s_1, ..., s_N$ are obtained using the MILP formulation described in Sections V-B or VI, and that QoC parameters (i.e., integrity enforcement rates $l_1, ..., l_N$) are either predefined, if MILP formulation from Section V-B is used (Scenario 1), or obtained using the MILP formulation from Section VI with respect to an upper utilization bound $\bar{U}_{\mathcal{M}}$ (referred to as Scenario 2). Thus, $\mathcal{M}$ is schedulable with non-preemptive EDF scheduler in the presence of non real-time traffic. However, unless the overall utilization $U_{\mathcal{M}}$ is 1, spare bandwidth will always be available at runtime. Then, the question is whether the remaining available bandwidth can be *opportunistically* used to further improve QoC by adding MACs to active sensor messages, and if so, how should the bandwidth be allocated to the messages?

Note that Scenario 2 captures situations where the system designer may use a lower value for overall utilization bound $\bar{U}_{\mathcal{M}}$ to reduce the size of MILP problem from (17), followed by opportunistic allocation of spare bandwidth among all sensors in the network; the bandwidth should be allocated based on the improvement to the overall QoC that any specific authenticated transmission would provide. For example, a QoC-optimal message set could be designed to utilize the network up to 95%, for which the MILP size, determined by the size of the time testing set, is not yet drastically affected by the utilization.

One of the main requirements for such opportunistic bandwidth allocation is that opportunistically adding authentication to transmitted sensor measurements should not affect schedulability of the initial real-time message set $\mathcal{M}$. Thus, for a message to receive an *opportunity* to extend its transmission during a network idle time, the idle time has to appear during a period when its normal (i.e., non-authenticated) transmission occurs and the duration of the idle time must be such that the message (once extended) can be completely transmitted prior to its deadline. Note that in order for a sensor to perform such analysis locally, it is only needed to know parameters of each message M_i since message requests occur at the beginning of each period p_i. Hence, such opportunistic bandwidth allocation will only add additional authentications to normal messages and thus the overall QoC guarantees will only be improved.

The main remaining challenge is the assignment of priorities to messages with additional MACs such that the improvement in the overall QoC guarantees is maximized. Note that in some protocols such as CAN, which is considered in the case study in Section VII, the message with highest priority will be transmitted and the protocol intrinsically resolves any conflicts. Our approach is to use a policy that maximizes the increase in the overall QoC by assigning the priority to these additional messages such that it corresponds to the improvement of the specific $\hat{\mathcal{J}}_i$. Specifically, consider opportunistically adding authentication to message M_i released at time instant t, where $t_{i_{k-1}} \leq t < t_{i_k}$, and $t_{i_{k-1}}$ and t_{i_k} are the closest preceding and following time instants when authenticated messages are to be released according to the initial complete real-time message set. We define $\Delta l_i(t)$ and the reward function $r_i(t)$ as

$$\Delta l_i(t) = \left\lfloor \frac{\min\left(t - t_{i_{k-1}}, t_{i_k} - t\right)}{p_i} \right\rfloor, \quad r_i(t) = \omega_i \hat{\mathcal{J}}_i(\Delta l_i(t)),$$

and assign priority to extend the message with MAC at time t to be equal to reward function $r_i(t)$. Intuitively, improvement in QoC from integrity enforcement closer to the middle between two scheduled periodic integrity enforcements is larger, than from immediate successive integrity enforcements followed by longer periods with no authentications before the next scheduled periodic enforcement. As we will show in Section VII, based on the above priorities, the network idle times can be fairly distributed over messages, so that the resulting integrity enforcements are intermittent, rather than periodic, which effectively further limits effects of attacks.

VII. Evaluation

To evaluate our approach for network scheduling and bandwidth allocation with QoC guarantees in the presence of attacks, we use a standard benchmark proposed by the Society of Automotive Engineers (SAE) [22]. This benchmark specifies communication requirements for automotive subsystems on an electric vehicle platform. Communication requirements consist of 53 messages between seven subsystems including the driver, braking system, transmission and vehicle control, battery and inverter/motor controllers, and the instrumentation cluster. Full message specifications are provided in Appendix A. Sporadic messages are not assigned minimum inter-arrival times in the benchmark specification. For our analysis, we assume that all sporadic messages are transmitted with $20\ ms$ period, and respective deadlines equal to their periods. All other messages are also assumed to have deadlines equal to their periods. Additionally, we will assume that the longest possible message is a full-length CAN message ($64\ bit$ payload with $533\ \mu s$ transmission time at $240\ kbps$).

We extend the benchmark by adding seven more messages ($54 - 60$ specified in Appendix A) that are necessary for realization of three additional control loops — cruise control, differential braking, and steering control for lane tracking, presented in detail in [16], [23], [24], respectively. We use available models of these three systems as an input to our reachability analysis framework presented in Section III to obtain QoC degradation curves. These curves and their piecewise linear approximations are shown in Fig. 8.

Vehicle model used for cruise control contains three states — deviation from desired distance to the preceding vehicle, deviation from desired speed, and acceleration. In steady state, all of these values are equal to zero, since the vehicle is moving at constant desired speed with correct distance from preceding vehicle. To determine maximal inter-enforcement distance l_1^{max}, we need to decide on the maximal error e_1^{max} that provides satisfactory system performance. In this work,

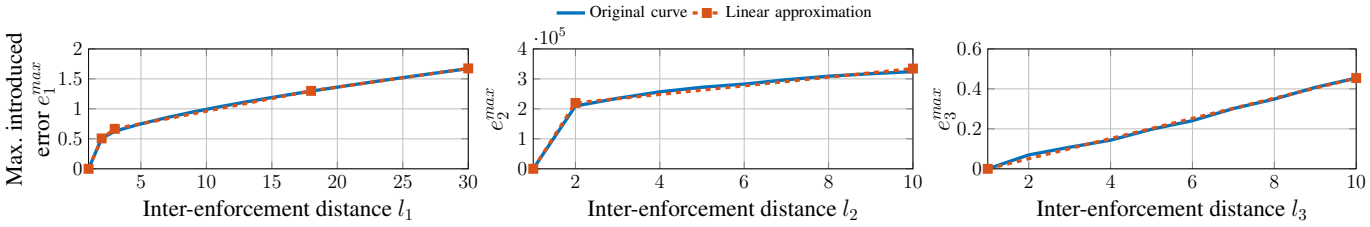

Fig. 8. QoC degradation (captured as the worst-case mean induced state estimation error) in the presence of attacks, with respect to integrity enforcement periods for: (left) Cruise control, (middle) Differential braking, and (right) Steering control for lane tracking.

we set allowed errors to be $0.1\ \frac{m}{s^2}$ on acceleration, $0.5\ \frac{m}{s}$ on speed and $1\ m$ on distance. This results in the norm of the mean estimation error of $e_1^{max} = 1.1225$, which mapped through the QoC degradation curve shown in Fig. 8(left) gives the maximum inter-enforcement distance of $l_1^{max} = 13$ sampling periods.

Differential braking model takes five states into consideration — brake pressure, lateral velocity, yaw rate of the vehicle, and inertial lateral position and velocity. Using parameters of the model from [23], we obtain the QoC degradation curve shown in Fig. 8(middle). This shows that the attacker has freedom to introduce an error that is several orders of magnitude larger than the noise, for any $l_2 > 1$. Hence, we consider $l_2^{max} = 1$.

Finally, steering control for lane tracking considers four states – lateral position error, lateral speed, yaw angle difference between vehicle and the road, and the speed at which this angle is changed. In steady state, when the vehicle is moving along a straight road, values of each of the states are zero, since the vehicle is holding center position. We assume that the vehicle is moving at speed of $30\frac{m}{s}$ and obtain additional model parameters as in [24]. Given these parameters, we obtain the QoC degradation curve shown in Fig. 8(right). Following the same methodology as for the cruise control system, we set allowed errors for lateral position error, lateral speed, steering angle, and angular velocity of axle as $0.2m$, $0.02\frac{m}{s}$, $0.18rad$, and $0.018\ \frac{rad}{s}$ respectively, which yields $e_3^{max} = 0.27$. This, in turn, maps to maximum inter-enforcement distance of $l_3^{max} = 6$ sampling periods.

We start by evaluating efficiency and scalability of our approach. To solve our MILP formulations we use Gurobi MILP solver [25]. We measure solver execution times on a platform with a 5^{th} gen. 3.0 GHz Intel i7 CPU and 16 GB of memory. If integrity is enforced on every data point for relevant subsystems in our message set, the link is overutilised with $U_{\mathcal{M}} = 1.0151$ (in case of CAN bus speed of 240 $kbps$). However, if we set the integrity enforcement rates to the predetermined maximum values ($l_1^{max} = 13$, $l_2^{max} = 1$, $l_3^{max} = 6$), utilization reduces to $U_{\mathcal{M}} = 0.9744$. Average solver execution time for the formulation with predefined QoC is $4.6123\ s$. Fig. 9 shows average MILP solver execution times for the same formulation and utilizations $0.1 - 0.9$ in increments of 0.1, as well as for 0.998 and 0.999. Optimizer execution time trends in case of optimal bandwidth allocation (presented in Section VI), are similar and are thus omitted. The increase in the problem size due to high utilization is visible, which

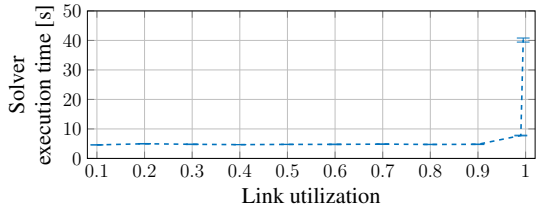

Fig. 9. Average MILP solver execution times and respective 95% confidence intervals for the modified SAE benchmark with utilizations $0.1 - 0.9$, as well as for 0.998, and 0.999. The execution time increases significantly due to increase in the problem size as $U_{\mathcal{M}}$ approaches 1.

supports our efforts on opportunistic bandwidth allocation. For a specific upper utilization bound of $\bar{U}_{\mathcal{M}} = 0.98$, we obtain $l_1^* = 5$, $l_2^* = 1$, $l_3^* = 4$, while the solver takes an average of $26.5978\ s$. Obtained utilization is $U_{\mathcal{M}}^* = 0.9791 < 0.98$.

Finally, we analyzed control performance for the these systems in the presence of attacks; the results are shown in Fig. 10 and Fig. 11. Fig. 10 (left) and Fig. 11 (left) confirm that system performance conforms to required minimum performance when $l_1 = l_1^{max} = 13$, $l_2 = l_2^{max} = 1$, and $l_3 = l_3^{max} = 6$, since the mean state errors do not exceed required limits. Fig. 10 (middle) and Fig. 11 (middle) illustrate the improvement in QoC when enforcement rates are optimized with respect to the upper utilization bound $\bar{U}_{\mathcal{M}} = 0.98$, resulting in $l_1^* = 5$, $l_2^* = 1$, $l_3^* = 4$. Finally, Fig. 10, Fig. 11 (right), show significant QoC improvement when respective control loops can opportunistically use available network idle times to authenticate sensor measurements, as proposed in Section VI-A, starting from the message set obtained from the optimization procedure for $\bar{U}_{\mathcal{M}} = 0.98$. In this case, average resulting authentication rates $l_1^{opport} = 1.64$, $l_2^{opport} = 1$, and $l_3^{opport} = 1.61$ are significantly higher than for the optimal allocation with $\bar{U}_{\mathcal{M}} = 0.98$.

Note that the attacker is assumed to have full knowledge on instants when both periodic and opportunistic authentication occur, and plans attacks accordingly. If opportunistic authentication points were unknown at attack design time, or impossible to predict, the attacker would eventually violate stealthiness conditions [12]. The final link utilization with idle times exhausted by opportunistic transmissions is 0.9972.

VIII. Conclusion

In this paper, we have presented a scheduling framework that jointly considers timing and security requirements for communication between sensors and controllers. We have shown how physics-aware QoC requirements can be translated

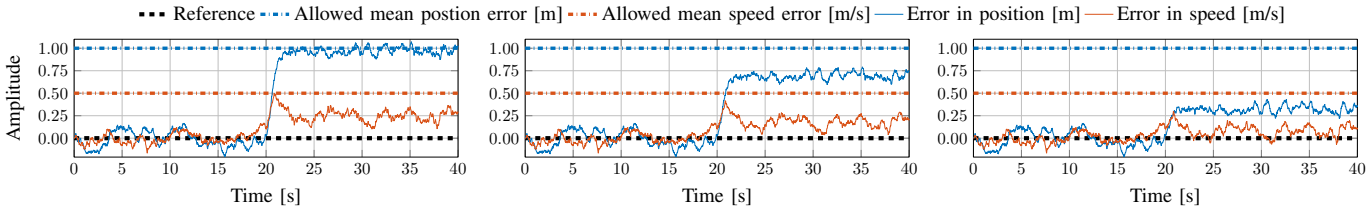

Fig. 10. Cruise control with three different integrity enforcement policies: $l_1^{max} = 13$ (left), $l_1^* = 5$ (middle), and variable l_1^{opport} that ranges from 1 to 5, with mean value of 1.64 (right). The attack begins at 20 s, and we present two states – deviation from desired distance and deviation from desired speed.

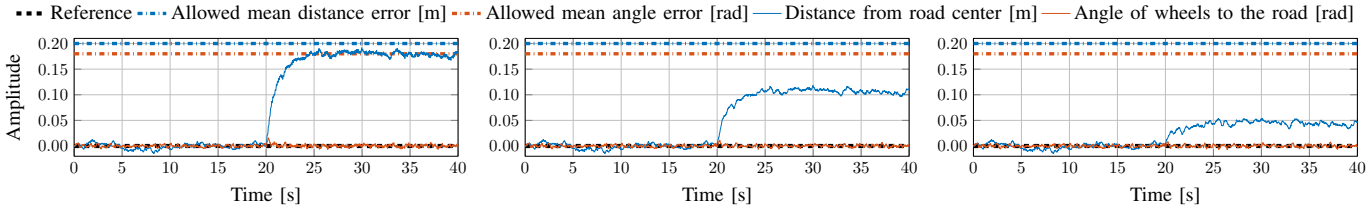

Fig. 11. Steering control for lane tracking with 3 integrity enforcement policies: $l_3^{max} = 6$ (left), $l_3^* = 4$ (middle), and variable l_3^{opport} ranging from 1 to 4, with mean value 1.61 (right). The attack begins at 20 s, and we present two states – distance from the road center and wheels angle relative to the road.

into real-time constraints, based on which an MILP problem can be formulated for QoC-aware bandwidth allocation. Additionally, we have shown how an MILP optimization could be used to maximize the overall QoC guarantees and ensure schedulability of non-preemptive sensor messages. Moreover, in cases where optimal bandwidth allocation may become inefficient (as network utilization approaches 1), we have provided an efficient runtime method for opportunistic bandwidth utilization in order to additionally improve QoC guarantees. Finally, we have demonstrated applicability of our framework on a standard automotive benchmark for CAN bus, and shown how an otherwise infeasible message set can be scheduled while ensuring that existing real-time guarantees are not violated, as well as satisfying QoC requirements.

As an avenue for future work we will extend this work to incorporate recent results on local (sensor-wise) authentication, as well as integrate the presented approach for scheduling of real-time messages with intermittent authentication with our work on scheduling security-aware real-time tasks [20].

REFERENCES

[1] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. of USENIX Security*, 2011.

[2] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway, Wired Magazine," 2015.

[3] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack," *GPS World*, vol. 23, no. 8, pp. 30–33, 2012.

[4] C.-W. Lin, Q. Zhu, C. Phung, and A. Sangiovanni-Vincentelli, "Security-aware mapping for CAN-based real-time distributed automotive systems," in *Int. Conf. on Computer-Aided Design (ICCAD)*, 2013, pp. 115–121.

[5] C.-W. Lin, B. Zheng, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware design methodology and optimization for automotive systems," *ACM Trans. on Des. Autom. of Elec. Syst.*, vol. 21, no. 1, p. 18, 2015.

[6] M. Hasan, S. Mohan, R. B. Bobba, and R. Pellizzoni, "Exploring opportunistic execution for integrating security into legacy hard real-time systems," in *IEEE RTSS*, 2016, pp. 123–134.

[7] T. Xie and X. Qin, "Improving security for periodic tasks in embedded systems through scheduling," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 3, Jul. 2007.

[8] M. Lin, L. Xu, L. T. Yang, X. Qin, N. Zheng, Z. Wu, and M. Qiu, "Static security optimization for real-time systems," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 22–37, Feb 2009.

[9] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *49th IEEE Conf. on Decision and Control (CDC)*, 2010, pp. 5967–5972.

[10] C. Kwon, W. Liu, and I. Hwang, "Analysis and design of stealthy cyber attacks on unmanned aerial systems," *Journal of Aerospace Information Systems*, vol. 1, no. 8, 2014.

[11] I. Jovanov and M. Pajic, "Relaxing integrity requirements for resilient control systems," in *56th IEEE Conference on Decision and Control (CDC)*, 2017.

[12] ——, "Relaxing integrity requirements for resilient control systems," *CoRR*, vol. abs/1707.02950, 2017.

[13] A. Anta and P. Tabuada, "On the benefits of relaxing the periodicity assumption for networked control systems over CAN," in *30th IEEE Real-Time Systems Symposium (RTSS)*, 2009, pp. 3–12.

[14] M. Pajic, I. Lee, and G. J. Pappas, "Attack-resilient state estimation for noisy dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 82–92, March 2017.

[15] R. Majumdar, I. Saha, and M. Zamani, "Synthesis of minimal-error control software," in *ACM EMSOFT*, 2012, pp. 123–132.

[16] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[17] K. M. Zuberi and K. G. Shin, "Scheduling messages on controller area network for real-time CIM applications," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 310–316, 1997.

[18] Q. Zheng and K. G. Shin, "On the ability of establishing real-time channels in point-to-point packet-switched networks," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1096–1105, Feb 1994.

[19] P. Belotti et.al, "On handling indicator constraints in mixed integer programming," *Computational Optimization and Applications*, vol. 65, no. 3, pp. 545–566, 2016.

[20] V. Lesi, I. Jovanov, and M. Pajic, "Security-aware scheduling of embedded control tasks," *ACM Trans. Embed. Comput Syst.*, 2017.

[21] A. Cervin, J. Eker, B. Bernhardsson, and K. Årzén, "Feedback–feedforward scheduling of control tasks," *Real-Time Systems*, vol. 23, no. 1-2, pp. 25–53, 2002.

[22] *Class C Application Requirement Considerations*, SAE J2056/1, 1994 SAE Handbook, Vol, 2, pp.23.366 - 23.272.

[23] T. Pilutti, G. Ulsoy, and D. Hrovat, "Vehicle steering intervention through differential braking," in *ACC*, vol. 3, 1995, pp. 1667–1671.

[24] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2014. [Online]. Available: http://www.gurobi.com

APPENDIX
MODIFIED SAE BENCHMARK SPECIFICATION

Message number	Message description	Normal transm. time [μs]	Extended transm. time [μs]	Period [ms]	Inter-enforcement distance [periods]	Source ECU	Destination ECU
1	Traction Battery Voltage	300	-	100	-	Battery	Vehicle Ctrl.
2	Traction Battery Current	300	-	100	-	Battery	Vehicle Ctrl.
3	Avg. Traction Battery Temp.	300	-	1000	-	Battery	Vehicle Ctrl.
4	Auxiliary Batter voltage	300	-	100	-	Battery	Vehicle Ctrl.
5	Max. Traction Battery Temp.	300	-	1000	-	Battery	Vehicle Ctrl.
6	Aux. Battery Current	300	-	100	-	Battery	Vehicle Ctrl.
8	Master Cylinder Pressure	300	-	5	-	Brakes	Vehicle Ctrl.
9	Line Pressure	300	-	5	-	Brakes	Vehicle Ctrl.
10	Transaxle Lubrication Press.	300	-	100	-	Transmission	Vehicle Ctrl.
11	Transaction Clutch Line Press.	300	-	5	-	Transmission	Vehicle Ctrl.
12	Vehicle Speed	300	433	20	13	Brakes	Vehicle Ctrl.
13	Traction Battery Gnd. Fault	300	-	1000	-	Battery	Vehicle Ctrl.
14	Hi&Lo Contactor Open/Close	300	-	20	-	Battery	Vehicle Ctrl.
18	Brake Switch	300	-	20	-	Brakes	Vehicle Ctrl.
21	Transmission Over Temp.	300	-	1000	-	Transmission	Vehicle Ctrl.
23	12V Power Ack. Vehicle Ctrl.	300	-	20	-	Battery	Vehicle Ctrl.
24	12V Power Ack. Inverter	300	-	20	-	Battery	Vehicle Ctrl.
25	12V Power Ack. Inv./Mot. Ctrl.	300	-	20	-	Battery	Vehicle Ctrl.
28	Interlock	300	-	20	-	Battery	Vehicle Ctrl.
29	Hi Contactor Control	300	-	10	-	Vehicle Ctrl.	Battery
30	Lo Contactor Control	300	-	10	-	Vehicle Ctrl.	Battery
31	Rev. and 2nd Gear Clutches	300	-	20	-	Vehicle Ctrl.	Transmission
32	Clutch Pressure Control	300	-	5	-	Vehicle Ctrl.	Battery
33	DC/DC Converter	300	-	1000	-	Vehicle Ctrl.	Battery
34	DC/DC Converter Current Ctrl.	300	-	20	-	Vehicle Ctrl.	Battery
35	12V Power Relay	300	-	20	-	Vehicle Ctrl.	Battery
36	Trac. Batt. Gnd. Fault. Test	300	-	1000	-	Vehicle Ctrl.	Brakes
37	Brake Solenoid	300	-	20	-	Vehicle Ctrl.	Brakes
38	Backup Alarm	300	-	20	-	Vehicle Ctrl.	Brakes
39	Warning Lights	300	-	20	-	Vehicle Ctrl.	Instr. Cluster
40	Key Switch	300	-	20	-	Vehicle Ctrl.	Inv./Motor Ctrl.
41	Main Contactor Close	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
42	Torque Command	300	-	5	-	Vehicle Ctrl.	Inv./Motor Ctrl.
43	Measured Torque	300	-	5	-	Inv./Motor Ctrl.	Vehicle Ctrl.
44	Forward/Reverse Command	300	-	20	-	Vehicle Ctrl.	Inv./Motor Ctrl.
45	Forward/Reverse Ack.	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
46	Idle	300	-	20	-	Vehicle Ctrl.	Inv./Motor Ctrl.
47	Inhibit	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
48	Shift in Progress	300	-	20	-	Vehicle Ctrl.	Inv./Motor Ctrl.
49	Processed Motor Speed	300	-	5	-	Inv./Motor Ctrl.	Vehicle Ctrl.
50	Inverter Temperature Status	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
51	Shutdown	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
52	Status/Malfunction	300	-	20	-	Inv./Motor Ctrl.	Vehicle Ctrl.
53	Main Contactor Ack.	300	-	20	-	Vehicle Ctrl.	Inv./Motor Ctrl.
54	Lateral Deviation	300	433	20	6	Motion Sensing	Vehicle Ctrl.
55	Lateral Deviation Rate	300	433	20	6	Motion Sensing	Vehicle Ctrl.
56	Yaw Angle Error	300	433	20	6	Brakes	Vehicle Ctrl.
57	Yaw Angle Error Rate	300	433	20	6	Motion Sensing	Vehicle Ctrl.
58	Inertial Lateral Position	300	433	20	1	Motion Sensing	Vehicle Ctrl.
59	Distance to Preceding Vehicle	300	433	20	13	Motion Sensing	Vehicle Ctrl.
60	Acceleration	300	433	20	13	Motion Sensing	Vehicle Ctrl.

TABLE I

MODIFIED SAE [22] BENCHMARK FOR CAN BUS. STANDARD BENCHMARK ENCOMPASSES 53 MESSAGES BETWEEN 7 SUBSYSTEMS. WE EXTEND THE BENCHMARK BY ADDING ANOTHER SUBSYSTEM, AND WITHOUT LOSS OF GENERALITY, WE OMIT MESSAGES FROM THE DRIVER ASSUMING CORRESPONDING SENSORS ARE LOCALLY CONNECTED TO THE VEHICLE CONTROLLER. NOTICE THAT ALL MESSAGES HAVE EQUAL TRANSMISSION TIME. THIS IS DUE TO TYPICAL HARDWARE LIMITATIONS THAT ALLOW ONLY A FULL BYTE TO BE TRANSMITTED. CONSEQUENTLY, EVEN THOUGH BASE LENGTH OF CERTAIN MESSAGES IS AS LOW AS ONE BIT, THEIR EFFECTIVE PAYLOAD IS ONE BYTE. NOTE THAT TO OBTAIN EXTENDED TRANSMISSION TIMES, A 32-BIT MESSAGE AUTHENTICATION CODE IS ADDED TO THE MESSAGE PAYLOAD.