Integrating Security in Resource-Constrained Cyber-Physical Systems

VUK LESI, ILIJA JOVANOV, and MIROSLAV PAJIC, Duke University

Defense mechanisms against network-level attacks are commonly based on the use of cryptographic techniques, such as lengthy message authentication codes (MAC) that provide data integrity guarantees. However, such mechanisms require significant resources (both computational and network bandwidth), which prevents their continuous use in resource-constrained cyber-physical systems (CPS). Recently, it was shown how physical properties of controlled systems can be exploited to relax these stringent requirements for systems where sensor measurements and actuator commands are transmitted over a potentially compromised network; specifically, that merely intermittent use of data authentication (i.e., at occasional time points during system execution), can still provide strong Quality-of-Control (QoC) guarantees even in the presence of false-data injection attacks, such as *Man-in-the-Middle* (MitM) attacks. Consequently, in this work, we focus on integrating security into existing resource-constrained CPS, in order to protect against MitM attacks on a system where a set of control tasks communicates over a real-time network with system sensors and actuators. We introduce a design-time methodology that incorporates requirements for QoC in the presence of attacks into end-to-end timing constraints for real-time control transactions, which include data acquisition and authentication, real-time network messages, and control tasks. This allows us to formulate a mixed integer linear programming-based method for direct synthesis of schedulable tasks and message parameters (i.e., deadlines and offsets) that do not violate timing requirements for the already deployed controllers, while adding a sufficient level of protection against network-based attacks; specifically, the synthesis method also provides suitable intermittent authentication policies that ensure the desired QoC levels under attack. To additionally reduce the security-related bandwidth overhead, we propose the use of cumulative message authentication at time instances when the integrity of messages from subsets of sensors should be ensured. Furthermore, we introduce a method for the opportunistic use of the remaining resources to further improve the overall QoC guarantees while ensuring system (i.e., task and message) schedulability. Finally, we demonstrate applicability and scalability of our methodology on synthetic automotive systems as well as a real-world automotive case-study.

CCS Concepts: • **Security and privacy** → **Distributed systems security**; *Intrusion detection systems*; • **Computer systems organization** → **Embedded and cyber-physical systems**; Embedded systems; • **Software and its engineering** → **Real-time schedulability**; • **Theory of computation** → *Linear programming*;

Additional Key Words and Phrases: Cyber-physical systems security, real-time scheduling, quality-of-control, mixed integer linear programming

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 agreement numbers N00014-17-1-2012 and N00014-17-1-2504.

Authors' addresses: V. Lesi, I. Jovanov, and M. Pajic, Duke University, Durham, North Carolina; emails: {vuk.lesi, ilija.jovanov, miroslav.pajic}@duke.edu.

2378-962X/2020/05-ART28 $15.00
https://doi.org/10.1145/3380866

ACM Reference format:
Vuk Lesi, Ilija Jovanov, and Miroslav Pajic. 2020. Integrating Security in Resource-Constrained Cyber-Physical Systems. *ACM Trans. Cyber-Phys. Syst.* 4, 3, Article 28 (May 2020), 27 pages.
https://doi.org/10.1145/3380866

1 INTRODUCTION

In this work, we focus on securing resource-constrained cyber-physical systems (CPS) from network-based false-data injection attacks over low-level networks used for real-time communication of control-related messages. With these *Man-in-the-Middle* (MitM) attacks, attackers can inject maliciously crafted data into communication between sensors and controllers, forcing a controlled physical plant into a potentially unsafe state; this is achieved either directly (by injecting false control commands) or through actions of the controller (if sensor measurements are falsified). Several such attacks have been reported recently (e.g., see Refs [8], [9], [18], and [20]); susceptibility of modern automotive systems to this type of attacks was illustrated in Refs [8] and [12]. These attacks are especially threatening as they enable a *remote* attacker to compromise safety-critical control features of a system, by taking over some of the components with access to a low-level safety-critical network used for control, before using them to transmit malicious control-related messages.

Protection against this type of attack is commonly based on data integrity enforcements using message authentication. Standard methods for ensuring authenticity of sensor data require the signing of message authentication codes (MACs) on the sensor electronic control units (ECUs), transmitting sensor measurements along with the MACs, and verification of the MACs at the controller ECUs. However, due to security-related overhead this approach may not be applicable to resource-constrained embedded platforms, which are especially dominant in legacy systems. For example, our experiments on a 96 *MHz* ARM Cortex-M3-based ECU show that executing a single-input-single-output PID controller update takes approximately 5 μs, while signing a 128 bit MAC over a single measurement requires around 100 μs. Thus, resource constraints may make it infeasible to provide continuous protection of sensing data by authenticating every transmitted sensor measurement. Consequently, in this work, we seek to answer the question of exactly how much security enforcement is sufficient and how we can exploit available system resources in order to improve the overall security guarantees, in terms of Quality-of-Control (QoC) in the presence of an attack.

Due to the recently reported security incidents, the problem of securing CPS has drawn significant attention, with research efforts focused on the impact of false-data injection attacks on system performance (mainly QoC), as well as the design of attack-detectors and attack-resilient controllers using a physical model of the system (e.g., see Refs [11], [25], [30], [32], [34], [39], and [43]). One of the main results is that even when physics-based intrusion detectors are used, by changing messages received at the controller from a subset of system sensors, an attacker could launch stealthy (i.e., non-detectable) attacks that force the plant into any undesired state through the actions of the controller [19, 26, 40].

On the other hand, we have recently shown how physical properties of the controlled system under consideration, can be exploited to relax integrity requirements for secure control [15–17]. Furthemore, by computing reachable regions of the state estimation error under stealthy attacks, control performance under attack can be evaluated for intermittent integrity enforcement policies—i.e., policies that only intermittently employ message authentication. In Ref. [22], we condense these reachable regions into *QoC degradation curves* that quantify the interplay between computational (and bandwidth) requirements imposed by security services and the QoC-guarantees under

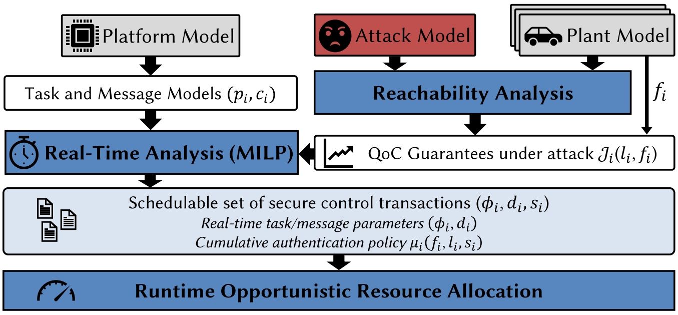

Platform Model
Attack Model
Plant Model
Task and Message Models (p_i, c_i)
Reachability Analysis
f_i
Real-Time Analysis (MILP)
QoC Guarantees under attack $\mathcal{J}_i(l_i, f_i)$
Schedulable set of secure control transactions (ϕ_i, d_i, s_i)
Real-time task/message parameters (ϕ_i, d_i)
Cumulative authentication policy $\mu_i(f_i, l_i, s_i)$
Runtime Opportunistic Resource Allocation

Fig. 1. Design-time methodology to integrate security in resource-constrained CPS; the use of cumulative intermittent message authentication policies enables tradeoffs between (i) required system resources (ensuring that all control functionalities perform within specifications even after "adding" security), and (ii) QoC guarantees under network-based false-data injection attacks on sensor measurements delivered to controllers.

attack. However, the use of such policies introduces new challenges for ensuring timeliness of deployed control functionalities, as the standard periodical task and message models under such relaxed integrity enforcement policies feature significant execution and transmission time variations. In Ref. [22], we only focus on the computational aspect of the problem and show how to guarantee timeliness for security-aware control tasks, while Ref. [21] presents our initial attempt to ensure timeliness of communication messages. Yet, Refs [21] and [22] only consider decoupled scenarios where the only concern for incorporating security is either ECU processing time (with the assumption that the network is not congested) *or* network bandwidth (while ECUs are not considered).

However, the problem of providing integrated QoC and security guarantees while ensuring timeliness in scenarios where both ECU processing time and network bandwidth are limited remains open, as both solutions from Refs [21] and [22] fall short in such case. Essentially, the methods from Ref. [22] for security-aware processor scheduling and Ref. [21] for security-aware network scheduling cannot be directly combined to obtain a solution for security-aware end-to-end scheduling, as the task/message models presented therein do not support precedence constraints between sensing, communication, and control computation. Moreover, Ref. [17] shows that block-authentication of sensor measurements has to be used for general types of dynamics of controlled physical processes, which results in workloads that cannot be modeled within the existing framework from Ref. [22].

Consequently, the main contribution of this work is a design-time methodology (Figure 1) that ensures that existing control functionalities will not be negatively affected by adding message authentication to enforce data integrity. Specifically, the methodology provides sensing-to-actuation timeliness guarantees for security-aware control that employs intermittent message authentication in order to guarantee that a desired QoC level is maintained even under attack. To capture the cases where block-authentication is needed while further reducing bandwidth requirements for the QoC guarantees under attack, we propose the use of intermittent cumulative authentication policies.

The presented methodology is enabled by the following additional contributions. First, we address modeling and capture schedulability conditions for security-aware control transactions; these transactions consist of *precedence-constrained, preemptive, real-time sensing tasks* that perform cumulative authentication, and security-aware *non-preemptive real-time messages* that support arbitrary offsets, and which are transmitted over a network with (dynamic or static) priority-based access. We show that existing schedulability conditions, based on the widely used

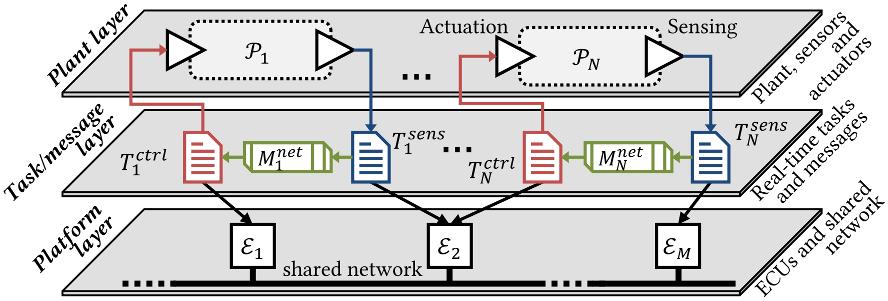


Fig. 2. System architecture with N physical plants ($\mathcal{P}_1, \ldots, \mathcal{P}_N$) that are sampled and controlled in real time by M ECUs ($\varepsilon_1, \ldots, \varepsilon_M$); the ECUs communicate with the corresponding plants' sensors and actuators over a real-time network, and the mapping of controllers for each plant $\mathcal{P}_i$ to a specific ECU ε_j is already performed.

requirements from Ref. [48] do not support general offsets, which prevents the use of our preliminary approach from Ref. [21]. Second, to compute a schedulable set of security-aware control transactions, we introduce a design-time synthesis method based on mixed integer linear programs (MILPs) as well as a platform architecture-based tradeoff analysis, which enables solving real-world size synthesis problems. Third, to further utilize resources available at runtime, we show that by opportunistically authenticating additional sensor measurements when computation time/bandwidth is available, we can further enhance QoC guarantees under attack. Finally, we show the use of our methodology on synthetic systems designed from automotive benchmarks guidelines, and an automotive case study.

This article is organized as follows. In Section 2, we present the system and attack models before introducing intermittent authentication policies for secure control of CPS (Section 3), and formalizing the end-to-end transaction modeling for secure control (Section 4). Schedulability analysis pertaining to the models is presented in Section 5, while Section 6 transforms the corresponding parameter synthesis problem into a MILP. Opportunistic use of remaining resources to improve the overall QoC guarantees under attack is presented in Section 7, before evaluating our approach in Section 8. Finally, Section 9 presents related work before concluding remarks in Section 10.

2 SYSTEM AND ATTACK MODEL

In this section, we present system architecture and model, including the attack model, and introduce cumulative authentication policies that ensure the desired QoC levels in the presence of attacks. We then formalize the problem of adding security guarantees against MitM attacks and outline our design-time methodology (shown in Figure 1) to integrate security in resource-constrained CPS.

2.1 System Architecture and Model without Attacks

We consider a common CPS architecture from Figure 2, where sensors for N physical plants $\mathcal{P}_i (i = 1, \ldots, N)$, as illustrated in the *plant layer* in Figure 2, communicate with plant controllers over a shared real-time network. We assume that each plant $\mathcal{P}_i$ can be modeled in the standard linear form as

$$\begin{aligned} \mathbf{x}_i[k+1] &= \mathbf{A}_i \mathbf{x}_i[k] + \mathbf{B}_i \mathbf{u}_i[k] + \mathbf{w}_i[k] \\ \mathbf{y}_i[k] &= \mathbf{C}_i \mathbf{x}_i[k] + \mathbf{v}_i[k], \end{aligned} \qquad (1)$$

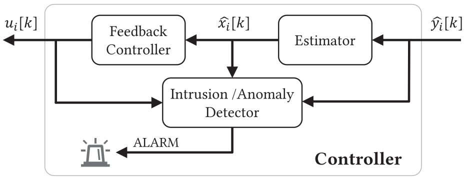


Fig. 3. General controller design. In addition to a standard estimator (i.e., observer) and a feedback controller, the controller employs a physics-based intrusion/anomaly detector.

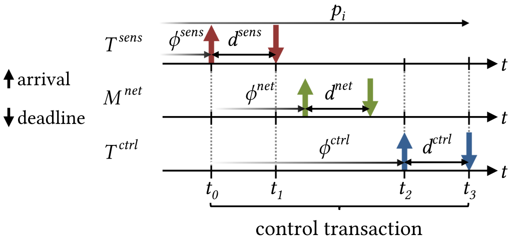


Fig. 4. Timing diagram of a control transaction—the precedence requirements for sensing (transmitting) task T_i^{sens}, message M_i^{net}, and control (receiving) task T_i^{ctrl} are captured by Constraints (2)–(4).

where $\mathbf{x}_i[k], \mathbf{y}_i[k]$, and $\mathbf{u}_i[k]$ denote the plant's state, output, and input vectors at time k, while $\mathbf{w}_i[k]$ and $\mathbf{v}_i[k]$ are process and measurement noise. In addition, each plant $\mathcal{P}_i$ is controlled by a feedback controller that, in the most general form, can be captured as

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

Here, $\mathbf{f}_i(\cdot)$ and $\mathbf{g}_i(\cdot)$ denote arbitrary linear mappings, which may, for example, describe an observer-based state feedback controller illustrated in Figure 3. In addition, $\hat{\mathbf{x}}_i[k]$ and $\hat{\mathbf{y}}_i[k]$ denote the estimate of the plant's state and sensor measurements received by the controller at time k. Also, as shown in Figure 3, we assume that each controller is equipped with a physics-based intrusion/anomaly detector that employs the plant model and a window of previous control inputs ($\mathbf{u}_i[k]$), state estimates ($\hat{\mathbf{x}}_i[k]$), and received sensor measurements ($\hat{\mathbf{y}}_i[k]$) to trigger alarms (e.g., as in Refs [15], [19], [25], [26], and [30]). For each controller, such detector is part of the controller's implementation, as illustrated in Figure 3, executing on the same ECU as the controller. Their executions do not introduce significant computational overheads, since state estimation is already performed for purposes of control, and relatively simple statistical manipulations of estimated differences between the expected and observed plant behaviors are not computationally intensive, as they involve computation of linear functions.

2.1.1 Task and Message Models. For each plant $\mathcal{P}_i$, measurement acquisition, packing and transmission is done by a periodic *sensing* (or *transmitting*) task denoted by T_i^{sens}. In addition, periodic *control* (or *receiving*) task T_i^{ctrl}, which may be executed on a different ECU, unpacks received measurements before using them for control updates in each sampling (i.e., actuation) period. Hence, the periods of these tasks are equal to the sampling period of the controlled plant—i.e., $p_i^{sens} = p_i^{ctrl} = p_i$. We also assume that mapping of tasks onto ECUs has already occured, as shown in Figure 2—i.e., the set $\mathcal{T}_{\mathcal{E}_j}, j = 1, \ldots, M$, of tasks executing on each of the M ECUs $\mathcal{E}_1, \ldots \mathcal{E}_M$ is known; for example, in the *platform layer* in Figure 2, the task set $\mathcal{T}_{\mathcal{E}_2}$ that contains T_1^{sens} and T_N^{ctrl} is mapped onto ECU $\mathcal{E}_2$. Thus, we assume that the worst-case execution times (WCET) for all these tasks are known, and let c_i^{ctrl} and c_i^{sens} denote the WCET on the assigned ECUs, for tasks T_i^{ctrl} and T_i^{sens}, $(i = 1, \ldots, N)$.

Each sensing task T_i^{sens} communicates sensor measurements to control task T_i^{ctrl} through a real-time message M_i^{net} with the same period p_i and the worst-case transmission time c_i^{net}, as illustrated in the task/message layer in Figure 2. Note that when no confusion arises, we refer to all T_i^{sens}, M_i^{net}, and T_i^{ctrl} as tasks. Finally, without loss of generality, we assume that actuation is done directly by control tasks, i.e., actuation commands are not transmitted as messages over the network, although the presented model can be easily generalized to cover this case.

Control Transactions. For any plant $\mathcal{P}_i$, we define a *control transaction* $\mathcal{T}_i$ as the chain of invocations of T_i^{sens}, M_i^{net}, and T_i^{ctrl} with all the tasks being precedence-constrained. Specifically, the earliest time a job of task T_i^{ctrl} may start execution is upon receiving the required sensor message. Similarly, network access for message M_i^{net} cannot be requested before task T_i^{sens} has prepared data for transmission. We capture these precedence constraints with non-zero offsets and constrained deadlines imposed on the tasks (Figure 4); we model the tasks in the standard $(WCET, period, offset, deadline)$ format as $T_i^{ctrl}(c_i^{ctrl}, p_i, \phi_i^{ctrl}, d_i^{ctrl})$, $M_i^{net}(c_i^{net}, p_i, \phi_i^{net}, d_i^{net})$, and $T_i^{sens}(c_i^{sens}, p_i, \phi_i^{sens}, d_i^{sens})$, with the precedence constraints specified as

$$\phi_i^{net} \geq \phi_i^{sens} + d_i^{sens}, \tag{2}$$

$$\phi_i^{ctrl} \geq \phi_i^{net} + d_i^{net}, \tag{3}$$

$$\phi_i^{ctrl} + d_i^{ctrl} \leq p_i, \tag{4}$$

and illustrated in Figure 4. To simplify our notation, Constraint (4) employs a standard assumption (e.g., as in Ref. [2]) that the delay between consecutive actuations for each plant $\mathcal{P}_i$ is bounded by the control period p_i; however, these constraints can be easily adjusted for any fixed sampling-to-actuation delay bounds that may be considered.

Finally, it is important to highlight that the period p_i and WCET c_i^{sens}, c_i^{net}, and c_i^{ctrl} are known and considered inputs to our design-time procedure, as we do not want to significantly affect the initial (i.e., non-secured) control deployment. On the other hand, to enforce the tasks' precedence, each control transaction imposes the aforementioned constraints between the offsets and deadlines used to model the transaction tasks. Yet, the actual values are **not** assigned *a priori*, i.e., the transaction set is considered incomplete, and our goal is to determine offsets and deadlines for all tasks that produce a schedulable set of control transactions even when security mechanisms are incorporated.

2.2 Attack Model

The considered system architecture is susceptible to network-based attacks, such as MitM attacks, on communication between sensors and controllers. The attacker can use actions of the controller to force the plant away from the desired state by injecting false data that differ from actual sensor measurements, consequently affecting the controller's estimation and thus the applied control inputs. To formally capture this, we use the standard attack model from [11], [26], [30], [31], and [43], where additional term $\mathbf{a}_i[k]$ captures the vector of values injected by the attacker at time k on compromised measurements—i.e., with MitM attacks, measurements received by the controller $\hat{\mathbf{y}}_i[k]$ may differ from the actual sensor measurements $\mathbf{y}_i[k]$. Specifically,

$$\hat{\mathbf{y}}_i[k] = \begin{cases} \mathbf{y}_i[k], & \text{without MitM attack} \\ \mathbf{y}_i[k] + \mathbf{a}_i[k], & \text{with MitM attack} \end{cases} \tag{5}$$

Due to attacks, the system evolution would not occur according to the model from Equation (1). Therefore, we differentiate system evolutions with and without attacks by adding superscript a to all variables affected by the attacker's influence. For example, we denote the plant's state and outputs when the system is under attack as $\mathbf{x}_i^a[k]$ and $\mathbf{y}_i^a[k]$, respectively. Hence, in the case of attacks, sensor measurements delivered to the controller can be modeled as

$$\hat{\mathbf{y}}_i^a[k] = \mathbf{y}_i^a[k] + \mathbf{a}_i[k] = \mathbf{C}_i\mathbf{x}_i^a[k] + \mathbf{v}_i^a[k] + \mathbf{a}_i[k]. \tag{6}$$

The attack vector $\mathbf{a}_i[k]$ is unknown and can have any value assigned by the attacker. The only constraint is that it may be sparse, depending on the set of compromised information flows from sensors to the controller; specifically, if communication from a sensor to the controller for plant $\mathcal{P}_i$ is not corrupted then the corresponding value in $\mathbf{a}_i[k]$ has to be equal to zero. Any assumptions about the set of compromised sensor flows (e.g., the number of the flows) can thus be captured by introducing constraints on the sparsity of the vector. However, unless stated otherwise, to simplify our presentation, we focus on the worst-case scenario, where the attacker is able to compromise all sensor flows for the plant once he/she decides to launch an attack.

With the use of standard cryptographic mechanisms, such as MACs, integrity of the received sensor data can be guaranteed, as we assume that the attacker does not have access to the shared secret keys used to generate the MACs. In addition, we assume that one of the attacker's goals is *to remain stealthy*, and, thus, in timesteps, when message authentication is used, the attacker cannot inject false data (i.e., $\mathbf{a}_i[k] = \mathbf{0}$) or the attack will be detected.[1] Furthermore, we assume that the attacker has unlimited computation power and full knowledge of the system, system architecture and plant models, as well as the time-points when authentication will be utilized. This allows him to plan ahead, and smartly craft false measurements to be injected over the network, such that they do not trigger the deployed detector, while deceiving the controller into pushing the plant away from the desired operating pointexamples of such attacks can be found in Refs [15], [17], [19], and [26].

Consequently, the attacker's goal is to maximally reduce control performance (i.e., QoC) while remaining stealthy—i.e., undetected by the system. Therefore, in addition to not inserting false data packets in time-frames when data authentication is enforced, the injected falsified sensor measurements should not trigger the anomaly/intrusion detection system employed at the controller.

3 DEFENDING AGAINST ATTACKS WITH INTERMITTENT DATA AUTHENTICATION

Enforcing data integrity for every communicated measurement packet may be infeasible due to additional computation associated with signing and verifying MACs, as well as additional bandwidth required to transmit them. For example, consider three sensing tasks executed on an ECU $\{T_1^{sens}(2,10), T_2^{sens}(2,10), T_3^{sens}(5,20)\}$—when a task is represented as $T(c,p)$, its offset is zero and relative deadline is equal to the period p. Let us assume that the security-induced computation overhead to sign measurements with a MAC is 2 time units. As shown in Figure 5(left), the new task set $\{T_1^{sens}(4,10), T_2^{sens}(4,10), T_3^{sens}(7,20)\}$ is infeasible; thus, even if the network can deal with the additional communication overhead, the transmitting ECU cannot authenticate (sign) every message.

On the one hand, a stealthy attack may significantly reduce QoC if the attacker has compromised a certain number of sensor flows (e.g., see Refs [19] and [30]). For any specific class of controllers from Figure 3, by injecting false sensor data that result in a skewed state estimation, the attacker deceives the controller to apply inappropriate control inputs that steer the plant away from the operating point. On the other hand, in Refs [15–17], we show how physical properties of a system can be exploited to relax integrity requirements for secure control of CPS. The idea is that the state estimation errors due to attacks have to increase slowly to avoid attack detection by the deployed physics-based detector from Figure 3. In addition, since each plant has its own dominant

[1] Note that the attacker, with access to the network, could launch Denial-of-Service attacks that prevent messages, including authenticated ones, from being successfully delivered to the controller. In this work, we do not consider such attacks since they are in general easier to detect in CPS with reliable communication networks.

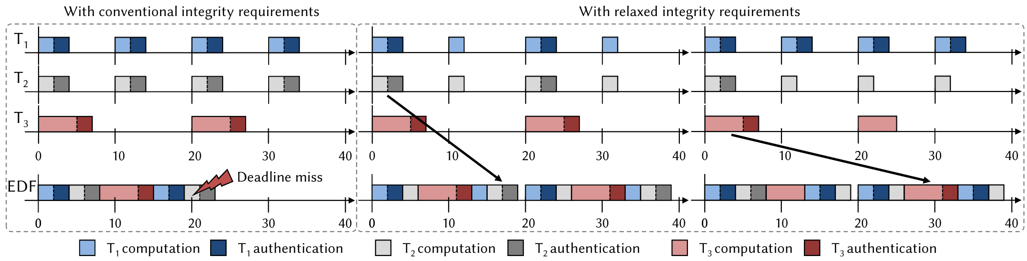


Fig. 5. Task set $T_1^{sens}(2, 10)$, $T_2^{sens}(2, 10)$, $T_3^{sens}(5, 20)$ is infeasible if overhead of signing sensor measurements is 2 time units in every sampling period (left). However, if T_1^{sens} and T_2^{sens} are allowed to authenticate every other period, and the initial authentication of T_2^{sens} is deferred until the second period, the task set is schedulable (center). On the other hand, if the goal is to maximize QoC guarantees for the first plant by always authenticating T_1^{sens} measurements, authentication rates for T_2^{sens} and T_3^{sens} can be reduced by authenticating every fourth and every other period, respectively, while still providing suitable QoC guarantees under attack, using the QoC degradation curves to guide formal tradeoff analysis (right).

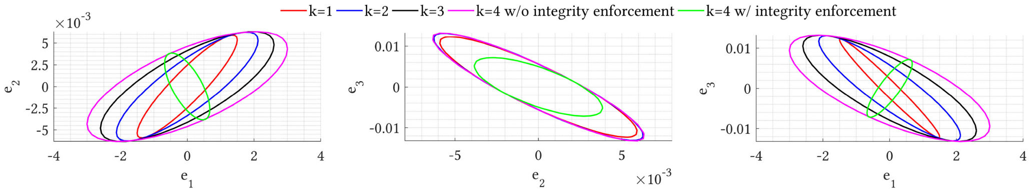


Fig. 6. State estimation error evolution due to a stealthy attack on distance sensing in an adaptive cruise control system—projections of the reachable regions in the three-dimensional state space (distance-speed-acceleration) are shown. Note that the attainable state estimation error is significantly reduced (but not zero) if integrity is enforced over every 4th measurement, while the regions grow infinitely without any integrity enforcement.

time-constant, which can be obtained by the plant model $\mathcal{P}_i$, in the presence of a stealthy attack, QoC can be significantly degraded only after some time has elapsed after the attack is launched.

QoC degradation under attack occurs due to errors in state estimation caused by the false-data injected at time-points when authentication is not used. Hence, for any data authentication policy, which can be captured as time-points where MACs are used (i.e., times k where $\mathbf{a}_i[k] = \mathbf{0}$), system performance under stealthy attacks can be evaluated by computing reachable regions of the state estimation error caused by the false data. Specifically, due to stealthy false-data injection attacks, the reachable regions $\mathcal{R}[k]$ and $\mathcal{R}$ of the state estimation error can be defined as [15–17]

$$\mathcal{R}[k] = \left\{ \mathbf{e} \in \mathbb{R}^n \;\middle|\; \begin{array}{c} \mathbf{e}\mathbf{e}^\intercal \preccurlyeq E[\mathbf{e}^a[k]]E[\mathbf{e}^a[k]]^\intercal + \gamma Cov(\mathbf{e}^a_k), \\ \mathbf{e}^a[k] = \mathbf{e}^a_k(\mathbf{a}_{1\dots k}), \mathbf{a}_{1\dots k} \in \mathcal{A}_k \end{array} \right\} \quad \text{and} \quad \mathcal{R} = \bigcup_{k=0}^{\infty} \mathcal{R}[k].$$

Here, $\mathcal{R}$ is the global reachable region of the state estimation error, while $\mathcal{A}_k$ denotes the set of all stealthy attacks $\mathbf{a}_{1\dots k} = [\mathbf{a}[1]^\intercal \dots \mathbf{a}[k]^\intercal]^\intercal$, and $\mathbf{e}^a_k(\mathbf{a}_{1\dots k})$ is the estimation error evolution due to the attack $\mathbf{a}_{1\dots k}$. Note that this general definition allows for the inclusion of additional information, such as the number and location of compromised sensors. Unless otherwise stated, we assume that measurements from all sensors are compromised when authentication is not used. For instance, Figure 6 shows the reachable regions of state estimation error due to stealthy attacks over the adaptive cruise control system described in Section 8.2 for the case with and without intermittent authentication.

In Ref. [22], we introduced a *QoC degradation curve* $\mathcal{J}_i(l)$ that, for any linear plant $\mathcal{P}_i$, directly quantifies the dependency between the security-induced computation and bandwidth overhead and the control performance (QoC) under attack, which is reduced due to the estimation errors. Each QoC degradation curve is computed numerically from the provided plant model, using the reachability analysis from Ref. [17] to derive the worst-case reachable region of the expected estimation error under attack (e.g., as in Figure 6),[2] for a range of authentication policy parameters (i.e., distances between authentications). Hence, $\mathcal{J}_i(l)$ can be used to bound QoC degradation as a function of l—the maximal time between consecutive MACs in data authentication policies; formally captured as

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

where $\mathcal{R}_i^l[k]$ denotes the reachable region $\mathcal{R}_i[k]$ computed for all data authentication policies with inter-authentication distance of l. Such QoC-degradation curves enable the designer to accurately adjust the system's working point by balancing between computational or network resource allocated for security and the returning QoC guarantees under attack, as the predefined QoC requirement can be directly mapped into security-induced overhead and vice versa.

To illustrate this, we revisit the example from Figure 5, and assume that for the first two plants, authenticating sensor measurements in every other sampling period ensures the desired QoC level in the presence of attack. Figure 5(center) shows that under such conditions, by deferring the initial authentication of T_2^{sens} until the second period, the task set becomes feasible. Note that, however, if every fourth measurement for the second plant and every other measurement of the third plant are authenticated, then the measurements for the first plant can continuously be authenticated, as in Figure 5(right). QoC degradation curves, computed for each plant independently, explicitly capture dependency between the required security-related overhead and control performance, and thus can be used to determine a suitable scenario with respect to the overall (for all plants) QoC guarantees—e.g., the overall QoC can be cast as a weighted sum of QoC degradation curves for all system plants.

3.1 Cumulative Data Authentication Policies

In general, depending on the considered plant's dynamics (i.e., matrices $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ in Equation (1)), it may not be sufficient to intermittently authenticate sensor measurements at one time point. Rather, integrity of f_i consecutive measurements should be ensured, with these time-windows appearing intermittently during system execution [17].[3] Implementing such data authentication policies with the use of standard MACs, where every authenticated message is signed with its own MAC added to the message, would require that f_i consecutive communication packets are extended to accommodate MACs. As the network is commonly a bottleneck in resource-constrained CPS, in this work, we propose the use of *cumulative message authentication* where a MAC is computed over several consecutive plant measurements, before being attached to the final message from the block; this significantly reduces the network load by transmitting a MAC for multiple consecutive data points as part of a single message [28, 35].

Therefore, we introduce the following definitions for cumulative data authentication policies that intermittently or periodically authenticate blocks of messages with sensor measurements.

[2] The estimation error under attack has a controllable (by the attacker) mean (i.e., $\mathbf{e}^a$) and a stochastic component with a covariance that follows from the measurement noise profile and plant dynamics, as in regular (e.g., Kalman) filters [17].

[3] As shown in Ref. [17], $f = \min(\psi, q_i^{un})$ with ψ being the observability index of the $(\mathbf{A}_i, \mathbf{C}_i)$ pair and q_i^{un} is the number of unstable eigenvalues of $\mathbf{A}_i$.

Definition 3.1. An intermittent cumulative data authentication policy $\mu_i = (\{t_j\}_{j=0}^{\infty}, f_i, l_i)$, with $t_{j-1} < t_j$ and $l_i = \sup_{j>0}(t_j - t_{j-1})$, ensures that $\mathbf{a}_{t_j} = \mathbf{a}_{t_j+1} = \cdots = \mathbf{a}_{t_j+f_i-1} = \mathbf{0}$, for all $j \geq 0$.

Definition 3.2. A periodic cumulative data authentication policy $\mu_i(s_i, f_i, l_i)$, where $0 \leq s_i \leq l_i - 1$, ensures that for all $j \geq 0$, $\mathbf{a}_i[s_i + l_i \cdot j] = \mathbf{a}_i[s_i + 1 + l_i \cdot j] = \cdots = \mathbf{a}_i[s_i + f_i - 1 + l_i \cdot j] = \mathbf{0}$.

Definition 3.1 imposes a maximum time of $l_i p_i$ (i.e., l_i control periods) between the initial authenticated measurements within blocks of f_i consecutive authenticated measurements. With periodic cumulative authentication policies from Definition 3.2, the time between initial authentications for consecutive blocks is always exactly $l_i p_i$, and authentication blocks start with the initial offset equal to $s_i p_i$.

A control transaction with an intermittent or periodic cumulative authentication policy applied to its tasks (resulting in security-related overheads) is referred to as a *secure control transaction*. For example, consider a secure transaction $\mathcal{T}_i$ from Figure 7, where a periodic cumulative data authentication policy $\mu_i(1, 2, 4)$ is implemented using cumulative MACs. During every four periods, overhead due to MAC signing for sensing task T_i^{sens} is spread over $f_i = 2$ jobs, while only one message M_i^{net} and job of T_i^{ctrl} include overhead due to authentication, and only after the last message from the authenticated block is prepared for transmission by T_i^{sens}.

Finally, the use of cumulative authentication introduces delay in verifying data integrity that has to be taken into account when QoC degradation curves are derived. Thus, in this case QoC degradation curves can be captured as $\mathcal{J}_i(l_i, f_i)$, which are computed from the plant model $\mathcal{P}_i$ using the reachability analysis we introduced in Ref. [16], as shown in the upper-right part of Figure 1. With our approach and the above definitions, only verification of each authentication code is delayed until the authentication block end—the use of received data for control *will not be delayed* since the controller is using even unsigned measurements (as in the initial non-secure deployment). While this ensures that there is no QoC degradation when the system is not under attack, as we show later, it still results in the desired QoC under attack. The reason is that the employed physics-based intrusion/anomaly detectors inspect each received measurement, effectively bounding the impact of stealthy attacks. Such attack impact, when the controller uses sensor measurements without waiting for the full block verification, is captured by the reachability analysis for the estimation errors.

Since the reachability analysis considers intermittent cumulative authentication policies from Definition 3.1, when used for periodic policies $\mu_i(s_i, f_i, l_i)$, as defined in Definition 3.2, it provides QoC guarantees **for any value** of s_i. For example, the QoC-degradation curves for adaptive cruise control, driveline management and lane keeping controllers, as functions of inter-authentication distance (l_i) and authentication block length (f_i), are shown in Figure 12. Note that the adaptive cruise control system requires that at least two consecutive measurements are authenticated (i.e., $f_{ACC} \geq 2$) due to the properties of the plant's dynamics.

QoC-degradation functions $\mathcal{J}_i(l_i, f_i)$ provide the basis for our analysis of tradeoffs between QoC guarantees under attack and the computational and network resources required for data authentication (i.e., security-related overhead). For each plant $\mathcal{P}_i$, the function $\mathcal{J}_i(l_i, f_i)$ is non-decreasing in l_i. In addition, the minimal required value for f_i can be directly computed from the model of $\mathcal{P}_i$ without significant QoC improvements being obtained by increasing f_i. Thus, the desired QoC requirements (e.g., a bound on $\mathcal{J}_i(l_i, f_i)$) can be directly mapped into constraints on the value of l_i, the number of non-authenticated communication packets between consecutive block authentications.

Remark 1. To simplify our presentation, in this work, we assume that when an attack is launched, measurements from all sensors transmitted over the network can be compromised;

therefore, when intermittent authentication is used for plant $\mathcal{P}_i$, then measurements of all the plant's sensors are authenticated. Yet, our results can be easily generalized for the case where only measurements from sets $\mathcal{K}_i^{at}$ are compromised and $\mathcal{K}_i^{auth}$ are authenticated, as the reachability analyses from Refs [16] and [17] directly cover these cases. This also facilitates the analysis of which sensors are more important to be protected (from the QoC under attack perspective), using recent methods from Refs [16], [17], and [46].

3.1.1 Overview of Our Approach. Our goal is to ensure the desired level of QoC for all controlled plants in resource-constrained CPS, even in the presence of network-based attacks. As resource constraints prevent continuous authentication of transmitted sensor measurements, we focus on *periodic* cumulative authentication policies, as for such block integrity enforcements are maximally spread apart. To achieve this, we propose the use of the design-time framework from Figure 1, which directly facilitates tradeoff analysis between the QoC guarantees under attack and security (i.e., authentication) overhead for ensuring intermittent integrity of sensor measurements. For each plant $\mathcal{P}_i$, $i = 1, \ldots, N$, the plant model and corresponding QoC curve $\mathcal{J}_i(l_i, f_i)$ are used to obtain constraints on employed periodic cumulative authentication policies, specifically, the values for l_i and f_i (but not s_i) that result in the desired QoC. In addition, from the platform model and the initial controller specification, regular (i.e., without overheads) and extended (i.e., including authentication) WCETs can be obtained, along with the control transaction period p_i.

On the other hand, for the task models to be complete and the intermittent authentication policies to be fully defined, it is necessary to derive feasible (i.e., schedulable) tasks' offsets and deadlines, as well as initial authentication offsets (s_i) for the cumulative authentication policies. Consequently, to allow for the execution of secure control transactions with the desired levels of QoC in the presence of attacks, in the rest of the article, we focus on the following scheduling problems.

Problem 1. *For a set of secure control transactions $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, complete the respective task/message sets and deployed periodic cumulative authentication policies, such that the obtained secure transaction set $\mathcal{T}$, mapped to available ECUs $\mathcal{E}_1, \ldots \mathcal{E}_M$, is schedulable under Earliest Deadline First (EDF) scheduler for ECUs and non-preemptive EDF for the network.*

Problem 2. *Starting from a schedulable set of secure control transactions $\mathcal{T}$, obtained from Problem 1, improve the overall QoC guarantees by utilizing remaining resources (ECU time, network bandwidth) with the use of intermittent cumulative data authentication policies.*

We consider EDF schedulers uniformly across ECUs and the network, since EDF is the optimal non-idle scheduler for preemptive task scheduling (i.e., on ECUs), while it outperforms rate-monotonic schedulers for realistic loads on non-preemptive networks such as Controller Area Network (CAN) [2, 48]. The main challenge in determining unknown parameters (task offsets, deadlines, and extended frame start times) is capturing schedulability conditions for preemptive-EDF on each ECU, as well as non-preemptive-EDF for the shared network. Thus, in the next section, we start by examining the mapping of the control- and security-related platform requirements into a security-aware control transaction model, which will provide a basis for our schedulability analysis and parameter synthesis procedure.

Remark 2 (Reduction of Control Rate vs. Reduction of Authentication Rate). The main idea behind this work is that with the simultaneous use of physics-based attack detection and cyber-based security mechanisms, such as message authentication, we will be able to provide strong QoC performance guarantees, even in resource-constrained CPS, in which it is not possible to protect the integrity of every transmitted sensor measurement. An alternative approach to the use of intermittent authentication would be to reduce the control rate to the levels that ensure that every transmitted sensor message can be authenticated. For instance, for our running example from

Figure 5, if control task rates are set to 20, 20, and 40 time units, respectively (instead of 10, 10, and 20), MACs can protect the integrity of every sensor measurement transmitted over the network. However, reducing the control rates (i.e., by increasing control task/sampling periods) results in a reduced control performance in the case without attacks, compared to the initial system that employs the nominal control periods. On the other hand, our goal is to add protection against network-based attacks with strong QoC guarantees in the presence of attacks, without negative effects on control performance (i.e., QoC) when the system is not under attack. With the use of intermittent authentication policies, this can be achieved by ensuring schedulability of the main control functionalities (tasks) at the nominal (i.e., initial) periods/rates, even when the authentication mechanisms are only intermittently utilized.

4 MODELING SECURE CONTROL TRANSACTIONS

Let us consider the workload imposed by a secure control transaction, such as the one shown in Figure 5(center, right). Schedulability analysis for such workloads using the standard task model ($WCET, period, deadline$) is highly pessimistic—clearly, the task sets from the figures would be rejected; the reason is that the standard task and message models accepting a single WCET parameter coarsely overapproximates the load on the ECUs and the shared network imposed by sparsely added security overhead. Thus, we need a model that captures the variable execution (or transmission) times of such security-aware real-time tasks.

The multi-frame task model [27] supports tasks that have execution times varying among consecutive invocations (called *frames*) in an arbitrary pattern. However, this model is overly general in that it allows any pattern of frames to be specified, and schedulability analyses for multi-frame tasks often assume that the worst-case alignment of frames is legal—exactly the scenario we want to avoid. In our case, it suffices to facilitate two frame sizes, regular and extended, with extended corresponding to executions that include security-related overhead, as well as additional parameters specifying extended frame period and offset; this allows for capturing of periodic cumulative data authentication policies, as the ones applied to tasks in Figure 5(center, right).

We develop a methodology for completing a set of transactions on the available shared network and ECUs, while taking into account the required level of periodic data integrity guarantees, obtained from the predefined QoC under attack requirements. Thus, we assume that non-zero task offsets and constrained deadlines are not known *a priori*. Instead, the respective task sets are considered incomplete in the sense that their periods and execution/transmission times are known, but the offsets and deadlines for each of the tasks that produce a schedulable set of transactions are to be determined. Hence, we model the *security-aware tasks* as $T_i(C_i, p_i, \phi_i, d_i, l_i, f_i, s_i)$, where

- $C_i = [c_i^{reg}, c_i^{ext}]$ is a WCET array for two frame types, regular and extended, respectively—c_i^{reg} is equal to c^{sens}, c^{net}, or c^{ctrl} for T_i^{sens}, M_i^{net}, and T_i^{ctrl}, respectively;
- p_i is the period at which jobs are released, ϕ_i is the release offset, and d_i is the task's deadline relative to its activation;
- l_i is the distance (i.e., number of control periods) between consecutive authentication blocks;
- f_i captures the length of the authentication block—i.e., the number of authenticated frames within one authentication period (i.e., within every interval of length $l_i p_i$);
- s_i is the initial authentication offset—i.e., the integer multiple of periods by which the initial authentication is deferred.

Note that the task offset consists of two components: ϕ_i and $s_i p_i$; ϕ_i is required to encode precedence constraints and applies to all jobs of the considered i^{th} task. On the other hand, $s_i p_i$ determines the additional offset of only extended frames, which provides a degree of freedom during

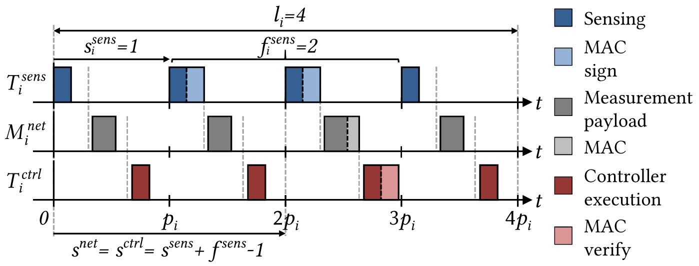


Fig. 7. Example of a secure control transaction when the periodic cumulative data authentication policy $\mu_i(1,2,4)$ is used. Note that only the transmitting task is extended for f_i consecutive invocations to perform cumulative authentication. On the other hand, the network message is extended only once, and, accordingly, the receiving task performs authentication (i.e., verifies the received MAC) once after receiving that measurement.

scheduling to avoid extended frame alignment scenarios emphasized in the motivating example (Figure 5(left)).

For tasks in any secure control transaction $\mathcal{T}_i$, some of the above parameters (i.e., s_i, f_i, l_i) directly follow from the employed authentication policy $\mu_i(s_i, f_i, l_i)$, as illustrated in Figure 7 for one example transaction. First, $l_i^{sens} = l_i^{net} = l_i^{ctrl} = l_i$, since the authentication period is the same for both tasks and the communication message. In addition, $f_i^{sens} = f_i$, as T_i^{sens} task computes a cumulative MAC over a block of f_i consecutive measurements, before attaching the MAC to the last message from the block. Also, $f_i^{ctrl} = 1$ since T_i^{ctrl} task verifies (i.e., authenticates) a block of consecutive measurements only once when it receives the cumulative MAC, prepared by T_i^{sens} and delivered by M_i^{net}. Thus, it also holds that $f_i^{net} = 1$.

Similarly, initial authentication offsets depend on the authentication policy used. First, $0 \leq s_i^{sens} \leq l_i - f_i$ since the first computation of cumulative MAC within a block must be done early enough to allow for execution of f_i consecutive extended frames within l_i periods of T_i^{sens}. Additionally, the initial extended frames of the message M_i^{net} and control task T_i^{ctrl} have constrained start times as $s_i^{ctrl} = s_i^{net} = s_i^{sens} + f_i^{sens} - 1$, as T_i^{sens} task computes cumulative MAC over f_i^{sens} periods, followed by an authenticated transmission and an authenticating control task, as shown in Figure 7.

Problem 1 can now be reformulated around the synthesis of feasible deadlines $(d_i^{sens}, d_i^{net}, d_i^{ctrl})$, offsets $(\phi_i^{sens}, \phi_i^{net}, \phi_i^{ctrl})$, and initial authentication offsets $(s_i^{sens}, s_i^{net}, s_i^{ctrl})$ for all secure control transactions $\mathcal{T}_i, i = 1, \ldots, N$, such that the precedence constraints from Equations (2)–(4) are satisfied, and for which the obtained complete transaction set $\mathcal{T}$ is schedulable under preemptive EDF for ECUs and non-preemptive EDF for the network. Thus, the following section starts by deriving schedulability conditions for the presented task model under preemptive and non-preemptive EDF scheduling.

5 SCHEDULABILITY ANALYSIS FOR SECURE CONTROL TRANSACTIONS

5.1 Schedulability of Security-Aware Tasks

We consider a schedulability condition for the sensing and control tasks based on the *processor demand criterion* [4]. Note that the condition from Ref. [22] cannot be used as it does not support the use of cumulative periodic authentication on sensing tasks, as well as general offset and deadline values for tasks and messages in secure control transactions. Necessary and sufficient schedulability conditions for the general task model (i.e., with non-zero offsets and deadlines differing from periods) under the preemptive EDF scheduler are formulated in Refs [4] and [7], starting from the following.

Definition 5.1 ([4]). The demand function df_i of a standard task $T_i(c_i, p_i, \phi_i, d_i)$ on interval $[t_1, t_2]$ is $df_i(t_1, t_2) = \sum_{\alpha_{i,j} \geq t_1, \delta_{i,j} \leq t_2} c_i$, where c_i is the WCET of the i^{th} task, while $\alpha_{i,j}$ represents the time of the j^{th} job arrival, and $\delta_{i,j}$ its respective deadline.

Theorem 5.2 ([4]). *A task set $\{T_1(c_1, p_1, \phi_1, d_1), \ldots, T_{N_{\mathcal{E}_j}}(c_N, p_N, \phi_N, d_N)\}$ is schedulable by preemptive EDF if and only if $\sum_{i=1}^{N_{\mathcal{E}_j}} df_i(t_1, t_2) \leq t_2 - t_1$, for all t_1, t_2 such that $t_1 < t_2$, where $N_{\mathcal{E}_j}$ is the number of tasks on the j^{th} ECU.*

Since, by definition, the demand function is piecewise constant, increasing in steps at time instants of job deadlines, the condition in Theorem 5.2 can be evaluated over a discrete and bounded time testing set. Formally, it is necessary to test the processor demand condition for all $t_{k_1} < t_{k_2} \leq t^{max}$,

$$\begin{aligned} t_{k_1} \in TS_{arr} &= \bigcup_{i=1}^{N} \{t | t = \phi_i + k_1 p_i, k_1 \in \mathbb{N}_0, t \leq t^{max}\}, \\ t_{k_2} \in TS_{dead} &= \bigcup_{i=1}^{N} \{t | t = d_i + k_2 p_i, k_2 \in \mathbb{N}_0, t \leq t^{max}\}, \end{aligned} \tag{7}$$

where $t^{max} = \max_i \phi_i + \max_i d_i + 2 \cdot lcm\{p_1, \ldots, p_N\}$ is the maximal time up to which the CPU demand has to be tested to ensure correctness of analysis [23], and lcm is the least common multiple.

We use this schedulability condition for schedulability analysis of security-aware T_i^{sens} and T_i^{ctrl} tasks—to simplify notation, we omit superscripts and denote the tasks as T_i where possible. To evaluate the demand function on interval $[t_{k_1}, t_{k_2})$, we compute the number of regular and extended frames released at or after t_{k_1}, which have deadlines at or before t_{k_2} as

$$\eta_i^{r\&e}(t_{k_1}, t_{k_2}) = max\left\{0, \left\lfloor \frac{t_{k_2} - \phi_i - d_i}{p_i} \right\rfloor - max\left\{0, \left\lceil \frac{t_{k_1} - \phi_i}{p_i} \right\rceil\right\} + 1\right\}. \tag{8}$$

Similarly, extended frames in this interval can be counted as

$$\eta_i^{ext}(t_{k_1}, t_{k_2}) = \sum_{m=0}^{f_i - 1} max\left\{0, \left\lfloor \frac{t_{k_2} - (s_i + m)p_i - \phi_i - d_i}{l_i p_i} \right\rfloor - max\left\{0, \left\lceil \frac{t_{k_1} - (s_i + m)p_i - \phi_i}{l_i p_i} \right\rceil\right\} + 1\right\}. \tag{9}$$

Here, the appropriate values for f_i should be used—i.e., $f_i^{ctrl} = 1$ for T_i^{ctrl} and $f_i^{sens} = f_i$ for T_i^{sens}.

The demand function for a single task can now be posed as the total processor demand of regular and extended frames as

$$df_i(t_{k_1}, t_{k_2}) = c_i^{reg} \eta_i^{r\&p}(t_{k_1}, t_{k_2}) + \Delta c_i \eta_i^{ext}(t_{k_1}, t_{k_2}), \text{ where } \Delta c_i = c_i^{ext} - c_i^{reg}. \tag{10}$$

We can thus formulate the necessary and sufficient schedulability condition as

$$\forall t_{k_1} \in TS_{arr}, \forall t_{k_2} \in TS_{dead}, \sum_{i=1}^{N} df_i(t_{k_1}, t_{k_2}) \leq t_{k_2} - t_{k_1}, \text{if } t_{k_1} < t_{k_2}. \tag{11}$$

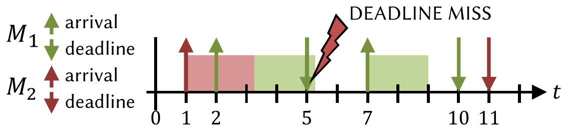


Fig. 8. Example message set $M_1(\phi_1 = 2, c_1 = 2, p_1 = 5, d_1 = 3), M_2(\phi_2 = 1, c_2 = 2.1, p_2 = 10, d_2 = 10)$—although the schedulability test for nonpreemptive messages with offsets from Ref. [48] is satisfied, M_1 misses its deadline at $t = 5$ due to an earlier release of message M_2.

5.2 Schedulability of Security-Aware Messages

To analyze schedulability of security-aware network messages (i.e., with periodic cumulative authentication), we start from the following theorem that provides a necessary and sufficient schedulability condition for *sporadic* real-time messages under non-preemptive EDF.

Theorem 5.3 ([47]). *A set of real-time messages $M_i(c_i, p_i, d_i)$, $1 \le i \le N$, where p_i is the minimum message inter-arrival time, is schedulable under non-preemptive EDF over a network shared with non-real-time messages with maximum transmission time c_{max}^{NRT} if and only if $\sum_{i=1}^{N} \frac{c_i}{p_i} \le 1$ and*

$$\sum_{i=1}^{N} max\left\{0, \left\lfloor \frac{t - d_i}{p_i} \right\rfloor + 1\right\} c_i + c_m \le t_k, \forall t_k \in TS, \tag{12}$$

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

To the best of our knowledge, there does not exist an efficient method to test schedulability for strictly periodic asynchronous messages under non-preemptive EDF. The conditions from Ref. [48] extend Theorem 5.3 for messages with offsets in order to support transaction scheduling. The resulting theorem from Ref. [48] replaces every appearance of relative deadline d_i in Theorem 5.3 with absolute deadline $d_i + \phi_i$ to account for offsets. In our case, using this theorem would be pessimistic since the conditions derived for sporadic messages cannot be adjusted for multi-frame messages. Also, examples as in Figure 8 show that the schedulability condition from Ref. [48] does not always hold.

On the other hand, a utilization-based test for non-preemptive EDF is derived in Ref. [3]. As our goal is to determine a set of offsets and deadlines that yields a schedulable set of secure transactions, this test cannot be used as it condenses all task properties into a single measure. Still, by following the reasoning presented therein, we formulate the following sufficient schedulability condition.

Theorem 5.4. *A message set $\{M_1(c_1, p_1, \phi_1, d_1), M_2(c_2, p_2, \phi_2, d_2), \ldots, M_N(c_N, p_N, \phi_N, d_N)\}$ is nonpreemptively schedulable by EDF if $\sum_i df_i(t_1, t_2) \le t_2 - t_1 - c_{max}$, for all t_1, t_2 such that $t_1 < t_2$, where $c_{max} = \max_i c_i$ is the longest of transmission times of all N messages.*

Proof. Suppose that the theorem's demand-based condition is satisfied for all t_1, t_2, and that there is a deadline miss at some instant $t_2^* = t_{dm}$. Let $t_1^* \le t_{dm}$ be the closest to t_{dm} instant such that the network is busy transmitting only those messages with deadlines $\le t_{dm}$. Then, right before t_1^*, the network may be idle or a message with deadline $\ge t_{dm}$ is being transmitted.

In the case when the network is idle right before t_1^*, then the total network demand imposed by all messages eligible to be transmitted during $[t_1^*, t_2^*]$ is $\sum_i df_i(t_1^*, t_2^*)$, by the definition of the demand function, and since there is a deadline miss at t_2^*, the demand must be greater than the network time available, i.e., $\sum_i df_i(t_1^*, t_2^*) > t_2^* - t_1^*$. This contradicts the theorem statement.

In the case when the network is transmitting a message with deadline $\geq t_{dm}$, then the worst-case network demand of all messages eligible to be transmitted during $[t_1^*, t_2^*]$ is $\sum_i df_i(t_1^*, t_2^*) + c_{max}$. Since there is a deadline miss at t_2^*, the demand must be greater than the available network time, i.e., $\sum_i df_i(t_1^*, t_2^*) > t_2^* - t_1^* - c_{max}$, which contradicts the theorem, and thus concludes the proof. □

The intuition behind this theorem can be supported by the claim that non-preemptive EDF schedules by time $t^* + c_{max}$ at least as much work imposed by a set of tasks as preemptive EDF schedules by t^* [3]. In this case, the total network demand by a security-aware message can be expressed as in Equation (10), with $f_i^{net} = 1$ used for extended transmissions in Equation (9). In addition, the time testing sets remain the same as in Equation (7). As we demonstrate on examples in Section 8.1, this condition is less conservative in cases when message transmission times are significantly shorter than their respective periods. We then show in Section 8.2 that this is commonly true in practical systems.

Remark 3 (Accounting for Jitter). To understand how realistic implementation phenomena such as jitter affect the presented analysis, we consider their effects on task and message scheduling. In the case of task-level jitter, existing approaches to jitter accounting can be applied [42]. In essence, if a task experiences jitter j_i, the inter-arrival spacing may be shorter than p_i. From the worst-case schedulability standpoint, this scenario pertains to the arrival pattern where all tasks arrive such that they must complete execution by the relative deadline $d_i - j_i$, rather than by d_i time units. Shortening the permissible deadline by the worst-case jitter can be easily included in the demand-based Condition (11). This does not affect the complexity of the MILP implementation of the parameter synthesis problem, as worst-case jitter figures as a set of known constant parameters. For message scheduling, in most cases, we do not need to use this approach, as the c_{max} term introduced in the non-preemptive schedulability conditions to account for the worst-case blocking any message may experience upon arrival, is rarely needed in its entirety; this holds since worst-case blocking will rarely occur. This conservativeness effectively captures jitter, as jitter levels are highly unlikely to exceed message transmission times in any practical network realization.

6 SYNTHESIS OF SCHEDULABLE SECURE CONTROL TRANSACTIONS

The schedulability conditions from Section 5, along with the task-precedence constraints from Section 2.1.1, can be used to formulate a parameter synthesis problem that produces a feasible set of task deadlines, offsets, and initial authentication offsets. However, non-linearity of functions counting the number of task invocations and message transmissions, Equations (8) and (9), precludes an efficient search of the parameter space. Thus, in this section, we map the demand-based schedulability conditions into a set of linear constraints, and formulate a MILP to synthesize task and message parameters that result in a schedulable set of secure control transactions. Since the schedulability conditions for preemptive and non-preemptive EDF differ only in the constant term c_{max} on the right side of the demand constraints from Theorems 5.2 and 5.4, in this section, we may omit superscripts *sens*, *ctrl*, and *net* for specific variables, where no confusion about the parameters arises.

Consider the workload of a sensing task T_i^{sens} that also incorporates cumulative periodical authentications. Let binary variables $a_{k,j,m}^i$ for T_i^{sens} indicate that the absolute deadline of the m^{th} extended frame of the j^{th} block of cumulative authentications is at or earlier than a time-testing instant t_k. This can be specified as

$$a_{k,j,m}^i = 1 \Leftrightarrow t_k \geq (s_i + m)p_i + \phi_i + d_i + (j-1)l_i p_i, \tag{13}$$

$$1 \leq i \leq N, \qquad 1 \leq k \leq |TS_{arr}| + |TS_{dead}|, \qquad 1 \leq j \leq \left\lfloor \frac{t^{max}}{l_i p_i} \right\rfloor, \qquad 0 \leq m \leq f_i - 1, \tag{14}$$

where TS_{arr} and TS_{dead} are defined in Equation (7). Note that control tasks T_i^{ctrl} and messages M_i^{net} are supported by simply removing the authentication iterator m (since $f_i^{ctrl} = f_i^{net} = 1$). A similar relation can be established for regular frames, where binary variables $b_{k,h}^i$ indicate that the h^{th} regular frame of the i^{th} sensing task is due by the k^{th} time testing instant t_k. This can be captured by

$$b_{k,h}^i = 1 \Leftrightarrow t_k \geq \phi_i + d_i + (h-1)p_i, \quad 1 \leq i \leq N, 1 \leq k \leq |TS_{arr}| + |TS_{dead}|, 1 \leq h \leq \left\lfloor \frac{t^{max}}{p_i} \right\rfloor. \tag{15}$$

Identical constraint can be written for control tasks T_i^{ctrl} and messages M_i^{net}. These variables enable us to concisely specify the number of respective jobs from Equations (8) and (9), respectively as

$$\eta_i^{r\&e}(t_{k_1}, t_{k_2}) = \sum_{j=1}^{\frac{t^{max}}{p_i}} \left(b_{k_2,h}^i - b_{k_1,h}^i\right), \tag{16}$$

$$\eta_i^{ext}(t_{k_1}, t_{k_2}) = \sum_{m=0}^{f_i-1} \sum_{j=1}^{\frac{t^{max}}{l_i p_i}} \left(a_{k_2,j,m}^i - a_{k_1,j,m}^i\right). \tag{17}$$

Hence, a task's processor demand can be cast as a linear function of variables $a_{k,j,m}^i$ and $b_{k,h}^i$ when Equation (16) and Equation (17) are instantiated in Equation (10). Note that since network and ECUs may not have the same hyperperiod, t^{max} should be computed independently for each ECU.

Since task offsets and deadlines are variables, the time testing instants are also variables, as defined in Equation (7). Thus, we should only consider the schedulability constraints from Theorems 5.2 and 5.4 for k_1 and k_2 such that $t_{k_1} < t_{k_2}$. This is achieved with constraint-enabling variables e_{k_1,k_2} such that

$$e_{k_1,k_2} = 1 \Rightarrow \sum_{i=1}^{N} df_i(t_{k_1}, t_{k_2}) \leq t_{k_2} - t_{k_1}, \tag{18}$$

for preemptive EDF, where e_{k_1,k_2} relates to the time testing instants as

$$e_{k_1,k_2} = 1 \Leftrightarrow t_{k_2} > t_{k_1}. \tag{19}$$

In addition, the far right side of Constraint (17) should be decremented by c_{max}^{net} when considering message scheduling, due to the scheduling non-preemptivity (Theorem 5.4).

Finally, to impose a bounded end-to-end delay, constraints that relate deadlines of tasks in a transaction can be specified as

$$d_i^{sens} + d_i^{net} + d_i^{ctrl} = p_i, \qquad 1 \leq i \leq N. \tag{20}$$

Remark 4 (Handling of Indicator Constraints). While the processor demand conditions can be directly implemented within an MILP, Constraints (13), (15), (17), and (18) cannot be directly specified as such in some MILP solvers. Those constraints can be linearized by using the "Big M" method for handling indicator constraints [5]. In the case of Constraints (13) and (15), we can write

$$-t_k + \phi_i + d_i + Ma_{k,j,m}^i \leq M - [s_i + m + (j-1)l_i]p_i, \quad -t_k + \phi_i + d_i + Mb_{k,h}^i \leq M - (h-1)p_i, \tag{21}$$

$$t_k - \phi_i - d_i - Ma_{k,j,m}^i < [s_i + m + (j-1)l_i]p_i, \qquad t_k - \phi_i - d_i - Mb_{k,h}^i < (h-1)p_i, \tag{22}$$

where M is a large constant. Similarly, Constraints (17) and (18) can be cast as linear constraints by enforcing

$$M(e_{k_1,k_2} - 1) + \sum_{i=1}^{N} df_i(t_{k_1}, t_{k_2}) \leq t_{k_2} - t_{k_1}, \tag{23}$$

$$t_{k_2} - t_{k_1} > M(e_{k_1,k_2} - 1), \qquad (23) \qquad t_{k_2} - t_{k_1} < Me_{k_1,k_2}. \tag{24}$$

Remark 5 (Handling of Strict Inequalities). Most MILP solvers do not allow specification of strict inequalities. Constraint (21) can be converted into non-strict inequalities by adding a small $\epsilon > 0$ to every t_k. Furthermore, Equation (23) can be directly converted into non-strict inequalities, while Equation (24) requires the addition of a small $\epsilon > 0$ on the left-hand side. Note that this may allow the time testing instants to meet during the solving process, i.e., $t_{k_1} = t_{k_2}$ is possible for some pair (k_1, k_2). This does not affect correctness of the formulation, but can only introduce redundant trivial demand constraints (i.e., over the interval of zero length). However, this does create an undesirable corner case. Despite the lack of an objective (recall that we are only interested in finding a feasible solution if such exists), solvers tend to minimize variables and may thus choose to zero all deadlines. This corner case is formally allowed if a time testing instant corresponding to a deadline of a task can coincide with its arrival. Since the demand constraint is satisfied (the processor demand over the interval of length zero is equal to the supply over the same interval), this modeling anomaly requires lower-bounding deadlines of each of the tasks. Simply, $d_i \geq 1$, for all i suffices.

Additionally, introducing ϵ to handle strict inequalities may affect the choice of value for M. Specifically, the values for M and ϵ must be selected such that no negative effects occur with the use of ”big-M” methods due to finite precision implementation of the employed MILP solver—that no constraint become active due to finite values for M. Thus, we set these values such that it holds that

$$M\delta_{int} + \delta_{constr} < \epsilon < 1 - M\delta_{int} - \delta_{constr},$$

where δ_{int} is the integer feasibility tolerance and δ_{constr} is the constraint satisfiability tolerance of the employed MILP solver. Moreover, M must be sufficiently large to ensure constraint satisfiability is not compromised for large t_k-s from the set TS.

The aforementioned constraints form an MILP formulation whose variables are the deadlines $(d_i^{sens}, d_i^{net}, d_i^{ctrl})$, offsets $(\phi_i^{sens}, \phi_i^{net}, \phi_i^{ctrl})$, and initial authentication offsets $(s_i^{sens}, s_i^{net}, s_i^{ctrl})$, as well as the introduced binary variables, but without an objective specification. If the feasible set of the problem is non-empty, our transaction set becomes complete and guaranteed schedulable. This approach, however, may be impractical for realistic scenarios (e.g., a unified MILP for the case study from Section 8.2 has over 10 million variables and 100 million constraints). Therefore, we now discuss methods for complexity reduction that we apply toward tackling realistic problems.

6.1 Complexity Reduction

To reduce the number of used variables and constraints, we first consider the time testing sets in Equation (7) for preemptive EDF. For a large number of arrival-deadline pairs (t_{k_1}, t_{k_2}), defining a variable indicating their ordering as in Equation (18) is not necessary, and thus the corresponding demand constraints can be omitted. For example, arrival time of any single job may never exceed the deadline of that, or any subsequent invocations of the task. Also, the deadline of a specific task invocation always occurs after the arrival of that or any earlier task invocations. Formally, since $e_{k_1,k_2} = 0, \forall i, \forall k_2 \geq k_1$ such that $\phi_i + k_1 p_i \geq d_i + k_2 p_i$, and $e_{k_1,k_2} = 1, \forall i, \forall k_2 \geq k_1$ such that $\phi_i + k_1 p_i < d_i + k_2 p_i$, Constraints (22)–(24) can be omitted. Similar relations can be drawn pairwise for every two tasks, given specific temporal parameters. This approach greatly reduces the number of used variables and constraints, especially for large hyperperiods.

A similar reasoning can be applied to variables $a^i_{k,j,m}$ that control the extended frame timing. Given specific temporal parameters of tasks, it is not necessary to encode the appearance of the j^{th} authentication block (i.e., the j^{th} sequence of m consecutive peak frames) for all instants in the time testing set, as suggested by Equations (13)–(14). This is true since we only seek to find a schedulable solution, which implies that the j^{th} authentication block must occur during the interval $[(j-1)l_ip_i, jl_ip_i]$, outside of which the value of $a^i_{k,j,m}$ is fixed and fully determined by tasks' temporal parameters. Formally, $(\forall i,j,k,m)\ \ (t_k > jl_ip_i \Rightarrow a^i_{k,j,m} = 0 \ and\ t_k < (j-1)l_ip_i \Rightarrow a^i_{k,j,m} = 1)$.

Similar holds for normal frames that must be scheduled within their respective periods:

$$(\forall i,k,h)\ \ (t_k > hp_i \Rightarrow b^i_{k,h} = 0 \ and\ t_k < (h-1)p_i \Rightarrow b^i_{k,h} = 1),$$

and thus the majority of Constraints (15) and corresponding variables $b^i_{k,h}$ that control normal frame timing can be eliminated. By enforcing these rules during problem encoding, the number of variables and constraints required to encode a realistic problem vastly reduces.

6.2 MILP Decomposition

Even with the discussed reductions in the number of variables and constraints, the presented MILPs may remain relatively complex for very large transaction sets. For these scenarios, we propose a decomposition approach that formulates the synthesis of schedulable secure control transactions as a sequence of MILPs, rather than a single program, since the schedulability tests from Section 5 can be decoupled between the ECUs and network. However, as we consider a parameter synthesis problem, rather than just a schedulability test, this decomposition is nontrivial—schedulable task parameters obtained for one part of the system do not guarantee feasibility of the remaining parts. In fact, the decomposition approach directly depends on the system architecture and its implementation.

6.2.1 Synchronous Sensing Platform Model. A commonly adopted platform model in offset-based scheduling of control transactions (as in Ref. [2]) assumes that all sensing tasks are initially released at the same time (i.e., $\forall i, \phi^{sens}_i = 0$ and $t_0 = n \cdot p_i$ for some n in Figure 4). In this case, in the first stage, we can run the ECUs' parameter synthesis MILP. Our objective could ensure that sensing tasks are scheduled as early as possible (minimized deadlines) while the opposite is desired for receiving tasks; i.e., they should execute as late as possible during their respective periods (maximized offsets) to ensure that the least conservative timing constraints are imposed on network messages (Figure 4). Trying to minimize all d^{sens}_i and maximize all ϕ^{ctrl}_i results in multivariate optimization that we solve by associating weights with each of the objectives (i.e., using blended objective). In the second stage, the network parameter synthesis MILP is formulated as a feasibility problem (without objective) searching for message offsets and deadlines that yield in a feasible transaction set.

Alternatively, in the first stage, we can run the network parameter synthesis MILP with the objective to maximize message offsets ϕ^{net}_i (which "leaves" time for transmitting tasks) and minimize deadlines d^{net}_i (which "leaves" time for receiving tasks). However, these objectives are conflicting, and since they have to be specified as a single blended objective function, heuristics can be used to adjust weights of individual message offsets and deadlines according to the execution times of sensing and control tasks (i.e., if the sensing task's WCET is longer than the control task's WCET, the message should be delayed more toward the end of the period). In the second stage, the ECUs' parameter synthesis MILP is formulated as a feasibility problem. However, there exist scenarios where this model is not the most accurate one; for instance, an ECU attached to multiple sensors may not necessarily have the capability to sample them instantaneously.

Consequently, our approach is that in the first stage, we execute the MILP formulation with lower complexity, which is better suited for this architecture, since that would reduce the time cost of reconfiguring task sets in the case that the MILP solver initially returns no solution.

6.2.2 Synchronous Network Access Platform Model. Another option is to assume that network access is synchronized—i.e., $\forall i, \phi_i^{net} = 0$ and $t_1 = n \cdot p_i$ for some n in Figure 4. In this case, the network MILP for parameter synthesis is executed first, with only message deadlines being subject to minimization to "leave" most time for sensing and control—resulting in the most efficient problem decomposition. On the other hand, if the ECUs MILP is run first, both sensing deadlines should be minimized and control offsets should be maximized as described in Section 6.2.1. Then, in the second stage, the ECUs' synthesis MILP is a feasibility problem, with additional simplifications since Constraints (2)–(4) become active (i.e., equalities hold), and for all i, d_i^{net} are pre-specified and $\phi_i^{net} = 0$. In terms of complexity, this approach is appropriate for large problems since it decouples the ECU and network analysis. Consequently, this reduces the number of variables and constraints per program since now only a part of the time testing instants remain variables.

In our evaluation in Section 8, the presented MILP decomposition approach is shown to be necessary for problems of realistic size. For example, Section 8.2 focuses on a realistic automotive case study whose full monolithic MILP formulation contains over 10 million variables and 100 million constraints, thus resulting in a program whose solving was unfeasible due the out-of-memory errors even on a workstation with 64 *GB* of RAM. While processing both the case study from Section 8.2 and synthetic systems from Section 8.1, when the decomposed approach was used, a reduction of at least one order of magnitude in terms of the variable/constraint count was observed. This, in turn, exponentially reduces the size of the underlying state space, resulting in fairly efficient MILPs for systems where solving their monolithic counterparts far exceeds available memory.

7 OPPORTUNISTIC AUTHENTICATIONS

The design-time framework from Section 6 addresses Problem 1, resulting in schedulable secure control transactions with the desired levels of QoC even in the presence of attacks. However, the overall QoC guarantees may be improved if the overall authentication rates, captured by l_i's, are increased; this can be achieved if additional system resources (ECU time, network bandwidth) are available. However, extending the presented MILPs by making the distances between authentications (i.e., l_is) to be variables, instead of predefined values obtained from the QoC requirements, does not scale. Hence, our methods from Refs [21] and [22] to optimally allocate resources in systems where only the network or only ECE scheduling is considered, cannot be employed for systems featuring many tasks/messages when both network and task scheduling are considered.

On the other hand, for secure transactions with periodic cumulative authentication policies $\mu_i(s_i, l_i, f_i)$ obtained by the MILP-based framework from Section 6, ECUs and the network will commonly not be entirely utilized at runtime. For instance, Figure 9 shows network traffic for the SAE benchmark [1], when critical messages are authenticated using our framework; here, vertical axis captures which message is transmitted over time. Transmission of message with message ID "0" indicates idle times that cannot be fully utilized using periodic integrity enforcement policies, but are otherwise available to further strengthen QoC in the presence of attack. Thus, in this section, we focus on how intermittent authentication can be added at runtime, on top of a system for which we already obtained strong timeliness and QoC-under-attack guarantees (i.e., Problem 2).

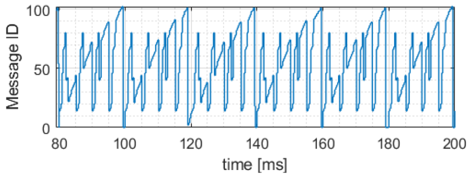


Fig. 9. Unauthenticated network traffic from Ref. [1], showing idle times (Message ID 0) that cannot be utilized with periodic policies.

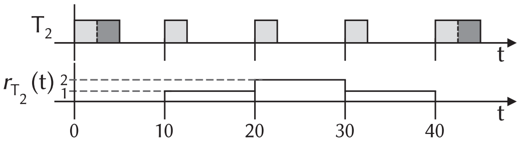


Fig. 10. Graphical representation of the reward function for opportunistic message authentications.

As our goal is to develop a runtime scheme that allocates available resources (CPU/network time) to authenticate additional sensor messages, we assume the following. First, each ECU needs to have the knowledge of the network's busy intervals, or equivalently, of the temporal parameters of the network's workload, to ensure that additional transmitted MACs do not affect timeliness of existing periodic traffic. This is a valid assumption in low-level control networks (e.g., CAN bus that is considered in the case study in Section 8.2), where traffic patterns are fully defined at design-time. Secondly, each ECU needs to have knowledge of its own available processing time to ensure that additional MAC signing or verification can be performed without violating timing constraints of existing transactions, and other periodic and worst-case sporadic workload. This is typically satisfied for constrained embedded platforms targeted by this general framework, as they commonly execute reservation-based Real-Time Operating System (RTOSs) that enforce runtime timeliness guarantees.

In such systems, our objective is to devise a runtime policy to determine optimal, or near-optimal *opportunities* for additional sensor measurements to be authenticated. Specifically, this policy defines, for each ECU, how to compute a priority level for any specific opportunistic MAC transmission; such priorities are then used to determine which message with opportunistically authenticated sensor measurements will be transmitted from all ECUs. Also, opportunistic authentications are only allowed outside the times used by the deployed periodic cumulative MAC policies $\mu(s_i, l_i, f_i)$. To improve the overall QoC guarantees, we consider QoC degradation curves $\mathcal{J}_i$ for every plant, and assign priority to a MAC transmission based on the level of improvement in the overall QoC that the specific opportunistically authenticated measurement would contribute—i.e., we assign such transmission priority at time t to be equal to the reward $r_i(t)$ that is defined as

$$r_i(t) = \omega_i \mathcal{J}_i(\Delta l_i(t), f_i), \text{ where } \Delta l_i(t) = \lfloor \min(t - t_{i_{k-1}}, t_{i_k} - t)/p_i \rfloor$$

Here, $t_{i_{k-1}}$ and t_{i_k}, such that $t_{i_{k-1}} \leq t \leq t_{i_k}$, are the nearest preceding and superseding periodic authentication release times from the policy $\mu(s_i, l_i, f_i)$. This ensures that additional authentications are favored in the middle of periods of regularly scheduled authentications, as they provide larger QoC improvements by imposing tighter attack constraints—i.e., the reward function $r_i(t)$ is increasing after the scheduled periodic authentication until the middle of the authentication period, after which it is decreasing as the next scheduled authentication approaches (Figure 10). Moreover, the weights ω_i enable boosting priority of more important plants (e.g., steering over climate control).

This approach is practical since the lightweight priority computation can be performed on the ECU itself in the case of the CAN bus, and the standard CAN protocol incorporates message priorities into the message identification field, while transmission conflicts are intrinsically resolved. Alternatively, the centralized scheduler assumed in TTCAN networks can enforce this policy, while each ECU in FlexRay networks features a *bus guardian*, which enforces design-time network access patterns at runtime and can be augmented with the aforementioned functionality. In Section 8.2,

Table 1. Distribution of Tasks and Messages Among Periods in Synthetic Workloads Used for Generic Evaluation, as Well as Non-QoC-Related Workloads Used for the Case Study; the Tasks and Messages were Obtained Using the Guidelines for Automotive Benchmarks from the SAE J2056/1 Standard

period [*ms*]	5	10	20	50	100	200	1000
share of preemptive (ECU) workload	2.5%	31.25%	31.25%	3.75%	25%	1.25%	5%
share of non-preemptive (CAN bus) workload	2.63%	32.89%	32.89%	3.95%	26.32%	1.32%	–

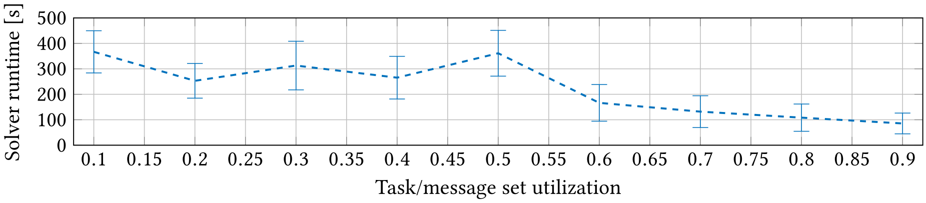


Fig. 11. Average Gurobi solver runtime and 95% confidence intervals for synthetic systems with utilizations 0.1 − 0.9, constructed by the guidelines for design of automotive benchmarks from the SAE J2056/1 standard.

we demonstrate how this approach can be used to significantly improve QoC under attack at runtime, at the expense of small amounts of utilized processing times and network bandwidth.

8 EVALUATION

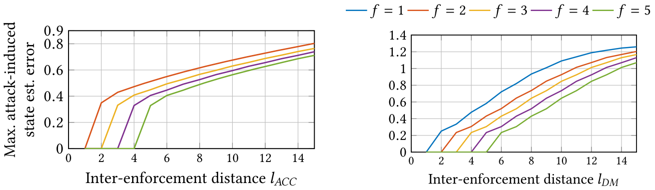
In this section, we evaluate our approach both on synthetic transaction sets (Section 8.1) and a realistic automotive case study (Section 8.2).

8.1 Evaluation on Synthetic Systems

For general evaluation, we generate over 5,000 synthetic systems, each featuring 10 to 50 control transactions, following the guidelines for the design of automotive benchmarks from SAE J2056/1 standard. Since the guidelines focus on defining ECU-bound workloads, we redistribute the angle-synchronous workload[4] and workloads with periods 1 *ms*, 2 *ms* evenly to workloads with other periods. This is done for synthetic message sets as most practical network workloads do not include messages with such short periods. Similar benchmark modifications were used in Ref. [10], and the resulting distribution among periods is summarized in Table 1. As in the SAE J2056/1 guidelines, we scale execution times to assess performance under different utilization levels. Message transmission times are computed based on full-size CAN bus payload of 64 *bits* by varying the transmission rate to vary network utilization. Finally, we randomly assign extended frame distances, and cumulative authentication block lengths in the range $l_i \in [1,5]$, $f_i \in [1,3]$, with 25%–50% of tasks/messages being QoC-related (and the remaining workload are standard real-time tasks/messages).

We evaluate scalability of our framework by applying the decomposed MILP approach to all synthetic systems to complete the generated transaction sets. Figure 11 summarizes Gurobi solver [29] runtime as a function of the number of tasks/messages and task/message set utilization,[5] showing applicability of our approach. Larger task sets typically cause longer solver runtime due to a generally larger parameter space. Relatively large variability can be attributed to random extended frame distances, which determine the hyperperiod and harmonicity of extended frame executions.

[4] Angle-synchronous tasks have periods that depend on the engine speed—i.e., the crankshaft angle determines job release.
[5] All computations are done on a Sandy Bridge EP-based workstation with dual 3.3 GHz Intel Xeon CPUs and 64GB of RAM.



(a) Adaptive cruise control (b) Driveline management (c) Lane keeping

Fig. 12. QoC degradation curves for three considered systems—maximal attack-induced state estimation error is bounded given a specific integrity enforcement policy determined by inter-enforcement distance l_i and authentication block length f_i. Note that the adaptive cruise control system requires at least two consecutive measurements to be authenticated ($f_{ACC}^{sens} \geq 2$).

Also, solver runtime is generally lower for unschedulable transaction sets regardless of the number of tasks since the solver is typically able to quickly prune large portions of the variable space, which expedites conclusions about unschedulability—average runtime in this case is 55 s.

8.2 Case Study

We consider a realistic automotive case study where controllers for adaptive cruise control, lateral control for lane tracking, and driveline management, are mapped onto three out of eight ECUs, with all ECUs also executing non-QoC-related workload as in Table 1. To model the controlled physical plants, we adopted physical system models from Refs [36], [37], and [38]. The control tasks are receiving sensor measurements from the eight ECUs communicating via a shared CAN bus. The network load consists of 70 full-sized CAN frames with period distribution specified in Table 1, and 8 full-sized CAN frames carrying sensor measurements with period $p_{ACC} = p_{LK} = p_{DM} = 20\ ms$. As 64 bit MACs are used to sign sensor measurements, to ensure low probability of forgery, an entire additional frame needs to be transmitted for an authenticated measurement, as the standard CAN payload is only 64 $bits$. With the standard 1 $Mbps$ CAN rate, regardless of ECU utilization, the system is not schedulable when every sensor measurement is signed.

Figure 12 shows QoC degradation curves for these systems, based on which we can map admissible levels of state estimation error due to attack into computation and bandwidth requirements. Specifically, we assume that state estimation error due to attack of no more than 0.4 m in distance to preceding vehicle, and no more than 0.1 $\frac{m}{s}$ in speed is allowed in the case of adaptive cruise control. Similarly, maximum attack-induced state estimation errors for lateral position error, its rate of change, yaw angle error, and its rate of change are set to 0.4 m, 0.1 $\frac{m}{s}$, 0.01 rad, and 0.01 $\frac{rad}{s}$, respectively. Finally, drive-shaft torsion and its rate of change state estimation errors due to attack are limited to 0.02 rad and 1 $\frac{rad}{s}$, respectively. Therefore, inter-enforcement distances and authentication block lengths resulting from these requirements are $l_{ACC} = 5, f_{ACC} = 3; l_{LK} = 10, f_{LK} = 2; l_{DM} = 10, f_{DM} = 1$.

Under these conditions, in the first step of our decomposed MILP approach, Gurobi solver takes an average of 2,716 s to return minimal deadlines for the considered message set and assign initial authentication start times such that timeliness can be guaranteed for network messages. In the second step, for a MILP that encompasses conditions for the three control ECUs, conditioned by the previously obtained message deadlines, Gurobi takes an average of 937 s to complete the secure transaction set with schedulable sensing task offsets and control task deadlines. Figures 13 and 14 show the resulting trajectories for adaptive cruise control and lane keeping systems when stealthy attacks start at $t = 20\ s$. Figures 13 and 14(left) show effects of the attack without authentication; both longitudinal and lateral control of the vehicle are entirely taken over by the stealthy attacker.

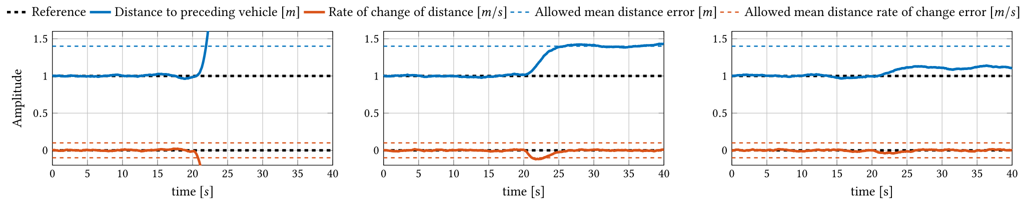


Fig. 13. Adaptive cruise control QoC under stealthy attack (starts at $t = 20$ s) without integrity enforcements (left), periodic cumulative authentication with $l_{ACC} = 5$ (center), and with intermittent cumulative authentication with $\hat{l}_{ACC} = 2.5$ (right).

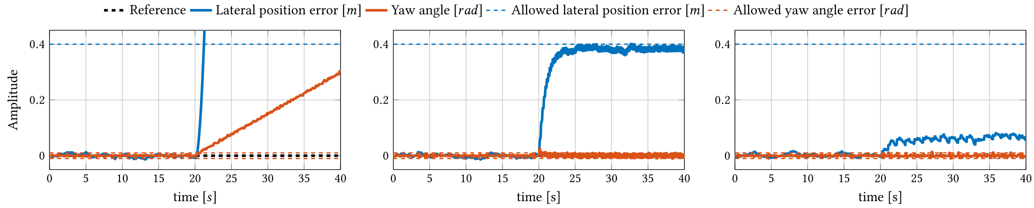


Fig. 14. Lane keeping QoC under stealthy attack (starts at $t = 20$ s) without integrity enforcements (left), with a periodic cumulative authentication with $l_{LK} = 10$ (center), and with intermittent cumulative authentication with $\hat{l}_{LK} = 2.86$ (right).

Figures 13 and 14(center) show how the attack impact is contained within permissible limits when integrity of sensor data is enforced with the aforementioned periodic cumulative policies, resulting in network utilization of $U_{net} = 0.68$.

To demonstrate benefits of using opportunistic scheduling to further improve the overall QoC under attack, we simulate additional sporadic network traffic as well as opportunistically add MACs (as described in Section 7) to sensor measurements that are not authenticated by periodic cumulative authentications. Sporadic messages are assumed to arrive with a minimum inter-arrival time of 10 *ms* utilizing up to 5% of the network bandwidth. The resulting mean inter-authentication distance for the three systems under consideration is $\hat{l}_{ACC} = 2.5$, $\hat{l}_{LK} = 2.86$, and $\hat{l}_{DM} = 2.31$, respectively. Figures 13 and 14(right) show significantly improved QoC levels under attack, while the shared network utilization increases on average by 10% due to opportunistic authentications. The final network utilization is $U_{net} = 0.84$. ECU utilization increases on average by only 1.5% to support signing and verification of additional MACs, illustrating the applicability of the presented framework.

9 RELATED WORK

Integrating security guarantees into legacy and resource-constrained systems has attracted significant research attention. In addition to Refs [21] and [22], which have been thoroughly discussed in Section 1, in Ref. [14], for example, the authors explore opportunistic execution of security services in legacy real-time systems, while leveraging hierarchical scheduling to ensure that schedulability of existing tasks is not impaired. The proposed security performance metric is the frequency of executions of security services. In Ref. [45], a novel scheduling policy is proposed for embedded systems to ensure schedulability of real-time control tasks subject to both timing and security constraints. This is achieved by optimal distribution of slack times that are computed after the schedulability of existing control tasks is guaranteed; and an optimal scheduler is constructed based on abstract relative security levels. In Ref. [24], the authors devise a security-aware EDF schedulability test with security services being grouped by a security level and execution of services from different groups is combined to increase Quality-of-Security (e.g., message encryption can be combined with authentication to protect both data confidentiality and integrity). Such group-based security model is integrated with EDF scheduling and a security-aware optimization problem is formulated around scheduling of suitable security services given a set of real-time tasks. However, no existing work provides a direct relationship between resource utilization and actual systems' performance pertaining to its main functionality (i.e., control performance, QoC)—in fact, only abstract *security levels* are considered.

Transaction scheduling is typically considered separately for time- and event-triggered communication models. For systems where network traffic patterns are determined by design, and resources (both computation power and network bandwidth) are severely constrained, timing constraints for transactions can be satisfied with careful offset/deadline enforcement—the approach considered in this article. Traditional offset-based rate monotonic schedulability analysis for distributed systems is presented in the original *analysis* framework from Ref. [44], and further improved in Refs [13] and [33]. Furthermore, this analysis is extended to EDF in Ref. [41]. However, only the standard task models are observed, mostly focusing on computing response times, while no optimization framework is devised to generate feasible offsets (or deadlines). In Ref. [6], the authors develop a technique to compute a (sufficient) region of admissible deadlines given a set of tasks under EDF, which facilitates optimization of a desired performance metric. Yet, this approach is non-trivial to integrate into an end-to-end schedulability analysis framework, due to its recursive algorithmic nature.

10 CONCLUSION

In this article, we have presented an MILP-based framework for integrating security guarantees with end-to-end timeliness requirements for control transactions in resource-constrained CPS. We have shown that the use of physics-based anomaly/intrusion detectors and intermittent message authentication results in strong QoC performance guarantees in the presence of network-based attacks without significant security-related resource overhead. We have also shown how the security-related overhead can be additionally reduced with the use of cumulative authentication policies, which can be implemented such that real-time guarantees for control-related tasks and messages are retained, while QoC in the presence of attacks is maintained within the permissible design-time limits. In addition, we have presented a method to integrate intermittent authentication policies in a near-optimal manner from the QoC standpoint, to opportunistically exploit available processor time and network bandwidth at runtime. As our approach fully supports cumulative authentication policies, it can be used for dynamical systems where solely authenticating a single sensor measurement periodically or intermittently is not sufficient to provide QoC guarantees under attack. Finally, for large-scale systems where a unified scheduling approach for all ECUs and network may be intractable, we have shown how the problem can be decomposed in a platform/implementation-specific manner. We have demonstrated scalability and effectiveness of our approach on both synthetic systems and a realistic automotive case study, and shown that security guarantees can be incorporated without violating existing timeliness properties even with limited resource availability.

REFERENCES

[1] 1994 SAE Handbook, Vol. 2, pp. 23.366–23.272. Class C Application Requirement Considerations.

[2] A. Anta and P. Tabuada. 2009. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. In *2009 30th IEEE Real-Time Systems Symposium*. 3–12.

[3] Sanjoy K. Baruah. 2006. The non-preemptive scheduling of periodic tasks upon multiprocessors. *Real-Time Systems* 32, 1–2 (2006), 9–20.

[4] Sanjoy K. Baruah, Louis E. Rosier, and Rodney R. Howell. 1990. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. *Real-Time Systems* 2, 4 (1 Nov. 1990), 301–324.

[5] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. 2016. On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications* 65, 3 (1 Dec. 2016), 545–566.

[6] E. Bini and G. Buttazzo. 2007. The space of EDF feasible deadlines. In *19th Euromicro Conference on Real-Time Systems (ECRTS'07).* 19–28.

[7] G. C. Buttazzo. 2011. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications* (3rd ed.). Springer, 110–114.

[8] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, and others. 2011. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of USENIX Security*, Vol. 4. IEEE, 447–462.

[9] Thomas M. Chen and Saeed Abu-Nimeh. 2011. Lessons from Stuxnet. *Computer* 44, 4 (2011), 91–93.

[10] P. Joshi, S. S. Ravi, Soheil Samii, Unmesh D. Bordoloi, and Sandeep Shukla, and Haibo Zeng. 2017. Offset assignment to signals for improving frame packing in CAN-FD. In *2017 IEEE Real-Time Systems Symposium (RTSS).* IEEE, 167–177.

[11] H. Fawzi, P. Tabuada, and S. Diggavi. 2014. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control* 59, 6 (2014), 1454–1467.

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

[13] J. C. Palencia Gutierrez, J. J. Gutierrez Garcia, and M. Gonzalez Harbour. 1998. Best-case analysis for improving the worst-case schedulability test for distributed hard real-time systems. In *10th EUROMICRO Workshop on Real-Time Systems (Cat. No.98EX168).* 35–44. DOI: https://doi.org/10.1109/EMWRTS.1998.684945

[14] M. Hasan, S. Mohan, R. B. Bobba, and R. Pellizzoni. 2016. Exploring opportunistic execution for integrating security into legacy hard real-time systems. In *2016 IEEE Real-Time Systems Symposium (RTSS).* 123–134.

[15] I. Jovanov and M. Pajic. 2017. Sporadic data integrity for secure state estimation. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC).* 163–169.

[16] I. Jovanov and M. Pajic. 2018. Secure state estimation with cumulative message authentication. In *2018 IEEE Conference on Decision and Control (CDC).* 2074–2079.

[17] I. Jovanov and M. Pajic. 2019. Relaxing integrity requirements for attack-resilient cyber-physical systems. *IEEE Transactions on Automatic Control* 64, 12 (2019), 4843–4858. DOI: https://doi.org/10.1109/TAC.2019.2898510 To appear.

[18] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, et al. 2010. Experimental security analysis of a modern automobile. In *IEEE Symp. on Security and Privacy (SP).* 447–462.

[19] Cheolhyeon Kwon, Weiyi Liu, and Inseok Hwang. 2014. Analysis and design of stealthy cyber attacks on unmanned aerial systems. *Journal of Aerospace Information Systems* 11, 8 (2014), 525–529.

[20] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE* 9, 3 (2011), 49–51.

[21] V. Lesi, I. Jovanov, and M. Pajic. 2017. Network scheduling for secure cyber-physical systems. In *2017 IEEE Real-Time Systems Symposium (RTSS).* 45–55. DOI: https://doi.org/10.1109/RTSS.2017.00012

[22] Vuk Lesi, Ilija Jovanov, and Miroslav Pajic. 2017. Security-aware scheduling of embedded control tasks. *ACM Transactions on Embedded Computing Systems* 16, 5s, Article 188 (Sept. 2017), 21 pages. DOI: https://doi.org/10.1145/3126518

[23] Joseph Y.-T. Leung and M.L. Merrill. 1980. A note on preemptive scheduling of periodic, real-time tasks. *Information Processing Letters* 11, 3 (1980), 115–118.

[24] M. Lin, L. Xu, L. T. Yang, X. Qin, N. Zheng, Z. Wu, and M. Qiu. 2009. Static security optimization for real-time systems. *IEEE Transactions on Industrial Informatics* 5, 1 (Feb. 2009), 22–37.

[25] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas. 2017. Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Transactions on Control of Network Systems* 4, 1 (March 2017), 106–117.

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

[27] A. K. Mok and D. Chen. 1997. A multiframe model for real-time tasks. *IEEE Transactions on Software Engineering* 23, 10 (1997), 635–645.

[28] Dennis K. Nilsson, Ulf E. Larson, and Erland Jonsson. 2008. Efficient in-vehicle delayed data authentication based on compound message authentication codes. In *68th IEEE Vehicular Technology Conference, VTC 2008-Fall.* 1–5.

[29] Gurobi Optimization Inc. 2014. Gurobi optimizer reference manual. (2014).

[30] M. Pajic, I. Lee, and G. J. Pappas. 2017. Attack-resilient state estimation for noisy dynamical systems. *IEEE Transactions on Control of Network Systems* 4, 1 (March 2017), 82–92. DOI: https://doi.org/10.1109/TCNS.2016.2607420

[31] M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G. J. Pappas, and I. Lee. 2017. Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators. *IEEE Control Systems* 37, 2 (April 2017), 66–81.

[32] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, Insup Lee, and G. J. Pappas. 2014. Robustness of attack-resilient state estimators. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS).* 163–174.

[33] J. C. Palencia and M. Gonzalez Harbour. 1998. Schedulability analysis for tasks with static and dynamic offsets. In *Proceedings 19th IEEE Real-Time Systems Symposium.* 26–37. DOI: https://doi.org/10.1109/REAL.1998.739728

[34] Fabio Pasqualetti, F. Dorfler, and Francesco Bullo. 2013. Attack detection and identification in cyber-physical systems. *IEEE Trans. Automat. Control* 58, 11 (2013), 2715–2729.

[35] A. Perrig, R. Canetti, J. D. Tygar, and Dawn Song. 2000. Efficient authentication and signing of multicast streams over lossy channels. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000.* 56–73.

[36] M. Pettersson. 1997. *Driveline Modeling and Control.* Ph.D. Dissertation. Linköping University.

[37] Jeroen Ploeg, Elham Semsar-Kazerooni, Guido Lijster, Nathan van de Wouw, and Henk Nijmeijer. 2014. Graceful degradation of cooperative adaptive cruise control. *IEEE Trans. on Intelligent Transportation Systems* 16, 1 (2014), 488–497.

[38] Rajesh Rajamani. 2011. *Vehicle Dynamics and Control.* Springer Science & Business Media.

[39] Yasser Shoukry, Michelle Chong, Masashi Wakaiki, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, Joao P. Hespanha, and Paulo Tabuada. 2018. SMT-based observer design for cyber-physical systems under sensor attacks. *ACM Transactions on Cyber-Physical Systems* 2, 1 (2018), 1–27.

[40] R.S. Smith. 2011. A decoupled feedback structure for covertly appropriating networked control systems. *Proceedings of IFAC World Congress* (2011), 90–95.

[41] Marco Spuri. 1996. *Analysis of Deadline Scheduled Real-time Systems.* Ph.D. Dissertation. Inria.

[42] John A. Stankovic, Krithi Ramamritham, and Marco Spuri. 1998. *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms.*

[43] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack models and scenarios for networked control systems. In *Int. Conf on High Confidence Networked Systems (HiCoNS).* 55–64.

[44] Ken Tindell. 1994. *Adding Time-offsets to Schedulability Analysis.* Citeseer.

[45] Tao Xie and Xiao Qin. 2007. Improving security for periodic tasks in embedded systems through scheduling. *ACM Trans. Embed. Comput. Syst.* 6, 3, Article 20 (July 2007), 20–es. DOI: 10.1145/1275986.1275992

[46] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti. 2016. Cross-layer codesign for secure cyber-physical systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 5 (May 2016), 699–711.

[47] Qin Zheng and K. G. Shin. 1994. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Transactions on Communications* 42, 234 (Feb 1994), 1096–1105.

[48] Khawar M. Zuberi and Kang G. Shin. 1997. Scheduling messages on controller area network for real-time CIM applications. *IEEE Transactions on Robotics and Automation* 13, 2 (1997), 310–316.

Received November 2018; revised October 2019; accepted January 2020